CENA 10.000 zł
ISSN 0867-3918 INDEKS 377112

64 PLUS 4 & AMIGA

6/'92

MIESIĘCZNIK UŻYTKOWNIKÓW KOMPUTERÓW COMMODORE

wakacje

wakacje

wakacje

wakacje



Drodzy Czytelnicy!

Na wstępie pragniemy przeprosić za znaczne opóźnienie ukazania się naszego amigowskiego zeszytu. Było ono spowodowane kilkakrotnym przesuwaniem terminu druku przez drukarnię. Aktualnie (tj. 20.06.) trwa już pakowanie i wysyłka - w najbliższym czasie powinny one dotrzeć do wszystkich zainteresowanych.

Kończy się rok szkolny - mamy nadzieję, że świadectwa będą wspaniałe! Nadchodzi czas zabawy i wypoczynku! Życzymy więc dużo słońca i pogody ducha!

Pamiętajcie: nasz następny numer ukaże się pod koniec lipca i będzie to wakacyjny numer PODWÓJNY!

REDAKCJA

Informujemy, że nasze pismo można w dalszym ciągu **zaprenumerować** - co daje pewność systematycznego otrzymywania (drogą pocztową). Nasz miesięcznik kosztuje w prenumeracie 10.000 zł. Prenumeratę można zawrzeć na okres nie krótszy niż dwa miesiące, w dowolnym okresie, maksymalnie do końca roku kalendarzowego. Wykupujący prenumeratę nie ponoszą kosztów przesyłki pocztowej.

Wadliwa dystrybucja „64 plus 4 & Amiga" przez przedsiębiorstwo RUCH jest przyczyną kłopotów, jakie mają czytelnicy chcący nabyć nasze pismo. Zdarza się, że otrzymujemy jako zwroty NIETKNIĘTE paczki zbiorcze. Tymczasem czytelnicy sygnalizują, że do wielu kiosków nie dociera ono wcale. Dlatego:

zapraszamy wszystkich chętnych do prowadzenia kolportażu „64 plus 4 & Amiga" (kluby, studia i sklepy komputerowe, księgarnie, osoby indywidualne itd.) do współpracy!

Oferujemy korzystne warunki!

Zainteresowanych prosimy o kontakt z działem dystrybucji pod adresem: Przedsiębiorstwo ABUK, 87-200 Wąbrzeźno, ul. 1 Maja 33.



miesięcznik nr 6(20)
czerwiec 1992
cena 1 egz.: 10.000

6 4 P L U S 4

WYDAWCA: ABUK Spółka z o.o.
REDAGUJĄ: Waldemar Szczygieł (redaktor naczelny) z zespołem.
ADRES REDAKCJI: Redakcja „64 plus 4", 85-166 Bydgoszcz 43, skrytka pocztowa 64.
OKŁADKA: Piotr Bartz.
SKŁAD: ABUK
DRUK: W.Z.G. Wąbrzeźno, okładka: Z.P. POLRASTER, Bydgoszcz.

OD REDAKCJI

W numerze :

NOWINKI

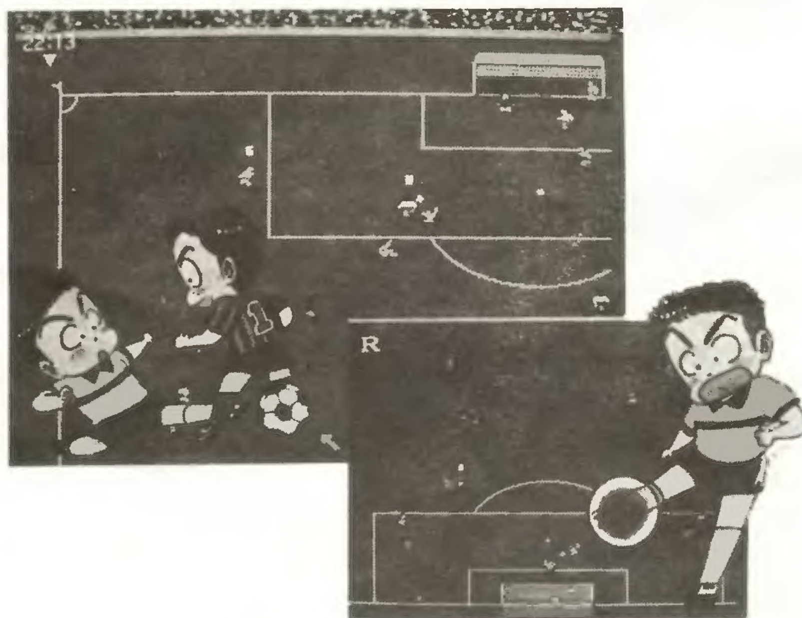


W CO POGRAMY???

W tym miesiącu trochę dla urozmaicenia napiszemy o gierkach.

- Firma TEAM 17 zaskoczyła nas ostatnio fantastyczną strzelaniną **Project X**. Project X to gierka bardzo dobrze dopracowana i właśnie taki pogląd głosi tą firma, że lepiej jest wypuścić mniej ale o szokującej jakości. Przygotowuje ona do wydania w czasie wakacji dwie kolejne wspaniałe gry:
- **SUPERFROG** - to wspaniała gra platformowa, z fantastycznie animowanymi postaciami o grafice zbliżonej do komiksów i filmów rysunkowych. Bardzo dobrze dobrane kolory oraz płynna animacja i dobra muzyka uzupełniają wszystko. Grafika będzie 32 kolorowa w kilku płaszczyznach. Na grę składa się 6 światów po cztery poziomy. Gra wymaga 1 mega pamięci. Przeznaczona dla graczy od lat 1 do 100.
- **CODENAME ASSASSIN** - to kolejna gra typu Hostages, Prince of Persia, etc. Bohater jest przywódcą podziemia, którego zadaniem jest zabicie (jak zwykle) złego wodza narodu. Bohater może się czołgać, biegać i skakać a jego celem jest przebycie drogi pełnej strażników, którzy rzucają nożami, strzelają i na dodatek szczują psami (zapewne pitbullterierami). Aby przez to wszystko przejść musi zbierać wszystko co mu "wlezie" pod nogi (za wyjątkiem strażników oczywiście!). Na zakończenie spotka on swojego największego przeciwnika, a potem bedzie już koniec. Grający musi być całkiem niezłym akrobatą komputerowym (co to oznacza to nie wiem ale tak napisali autorzy gierki!). Gra składa się z 832 (ośmiuset!!!) kolorowych lokacji, wymaga 1 megabajta pamięci i zajmuje tylko trzy dyski. Gra będzie opublikowana tylko na Amigę.
- **STRIKER** to nowy produkt firmy RAGE SOFTWARE, który zostanie wypuszczony niebawem. Jest to kolejna wersja gry Kick Off znanej i lubianej w szerokich kręgach, począwszy od graczy a skończywszy na poważnych ludziach zajmujących się pracą na komputerze. Nowość polega na ilości obiektów oraz możliwościach w definiowaniu parametrów gry ale przede wszystkim na tym, że zostały poprawione procedury odpowiadające za zachowanie się piłki i graczy. W grze są bardzo dobrze opracowane metody strzału i kiwania się z przeciwnikami. Przyciśnięcie raz przycisku w joysticku powoduje strzał w kierunku w jakim zwrócony jest joystick, natomiast przyciśnięcie dwa razy tego przycisku powoduje przytrzymanie piłki i umożliwia kiwanie się z przeciwnikami. Nie ma problemów z wykonaniem mocnych strzałow, jak to miało miejsce w poprzednich grach, że piłka była pozbawiona energii zanim osiągnęła swój cel. Piłka odbija się doskonale od innych przeciwników i zachowuje się całkiem "normalnie". Egzekwowanie rzutów rożnych i karnych jest usprawnione i można podkręcać piłkę oraz doskonale kontrolować siłę kopnięcia. Natomiast w przypadku obrony rzutu karnego, kontrola bramkarza została rozszerzona. Gra jest oceniana bardzo wysoko przez wszystkie czasopisma zachodnie i należy tylko czekać aż do nas dotrze.

Marcin "Duddie" Dudar



NIEŚMIERTELNI

C-16

W numerze 4/92 naszego pisma zamieściliśmy artykuł pt.: NIEZNISZCZALNI . Temat "nieśmiertelności" - jak się okazało zainteresował naszych czytelników. Dziś publikujemy kolejne dwa listy. Pierwszy nadesłał kol. Piotr z Będzina.

Jestem użytkownikiem C+4. Zaciekawiła mnie bardzo treść artykułu "Niezniszczalni". Ponieważ sam pracowałem nad ułatwieniami do gier przesyłam efekty mojej twórczości. Wprowadzenie tych ułatwień jest stosunkowo proste - po uruchomieniu gry przerywa się ją przez naciśnięcie klawisza RESET. Po dokonaniu zmian uruchamia się grę rozkazem SYS...

AARDVARK	POKE 9504,x	- ilość "żyć"
	POKE 9,666,100	- nie odejmuje "żyć"
	SYS 9458	
AURIGA	POKE 7686,x	- ilość "żyć"
	POKE 11302,100	- bezbronni przeciwnicy
	SYS 7768	
ASTRO PLOMBER	POKE 5435,234	- nie odejmuje "żyć"
	POKE 6632,0	- nie odejmuje "powietrza"
	POKE 6586,0	- nie odejmuje "paliwa"
	SYS 4100	
AUTO ZONE	POKE 7089,x	- ilość "żyć"
	POKE 9456,100	- nie odejmuje "żyć"
	SYS 4080	
BABY BERKS	POKE 8254,0	- nie odejmuje "żyć"
	SYS 5190	
BERKS III	POKE 9846,0	- nie odejmuje "żyć"
	SYS 4232	
BIG MAC	POKE 12988,100	- nie odejmuje "żyć"
	SYS 7000	
BREAK OUT	POKE 7866,0	- nie odejmuje "żyć"
	SYS 6368	
BLAZE	POKE 9734,x	- ilość "żyć"
	POKE 8789,100	- nie odejmuje "żyć"
	SYS 9727	
CHOPPER WAR	POKE 4522,49 do 57	- ilość "żyć"
	POKE 11202,100	- nie odejmuje "żyć"
	SYS 4447	
COPS & ROBBER	POKE 12406,x	- ilość "żyć"
	POKE 12843,100	- nie odejmuje "żyć"
	SYS 14031	
CUTHBERT COLLER	POKE 4413,49 do 57	- ilość "żyć"
	POKE 6650,100	- nie odejmuje "żyć"
	SYS 4352	
DIZASTER	POKE 14281,100	- nie odejmuje "żyć"
BLASTER	SYS 12288	
DANGER ZONE	POKE 6034,x	- ilość "żyć"
	POKE 6064,100	- nie odejmuje "żyć"
	SYS 5819	
EXORCIST	POKE 16216,14	
	POKE 16214,0	- nie odejmuje energii
	POKE 6876,3	- jesteś niewidzialny dla wroga
	SYS 4576	
GUN LAW	POKE 4381,x+1	- ilość "żyć"
	POKE 4629,100	- nie odejmuje "żyć"
	SYS 4135	
GUSZTAV	POKE 8015,x	- ilość "żyć"
	POKE 6647,100	- nie odejmuje "żyć"
	SYS 5376	
GALACTIK BATTLE	POKE 9835,100	stała temperatura
	SYS 4110	
GUZZLER	POKE 4128,x	- ilość "żyć"
	POKE 4133,x	- ilość "bomb"
	POKE 6454,100	- nie odejmuje "bomb"
	POKE 9027,100	- nie odejmuje "żyć"
	SYS 4108	
MAC STONEY	POKE 10212,100	- nie odejmuje "żyć"
	SYS 8951	
MONKEY MAGIC	POKE 10658,100	- nie odejmuje "żyć"
	POKE 7788,96	- jesteś niewidzialny dla wroga
	SYS 9264 + SHIFT	
MISSION MARS	POKE 6581,176 do 185	- ilość "żyć"
	POKE 4226,100	- nie odejmuje "żyć"
	SYS 8448	
ROBO KNIGHT	POKE 12171,0	- zatrzymanie upływu czasu
	POKE 15968,0	- nie odejmuje "żyć"
	SYS 4112	
ROCK MAN	POKE 8464,234	- zatrzymanie upływu czasu
	POKE 9757,x	- ilość "żyć"
	POKE 8596,234	- nie odejmuje "żyć"
	SYS 4102	
PETALS OF DOOM	POKE 15721,100	- nie odejmuje "żyć"
	SYS 15593	
SCRAMBLE	POKE 14977,x	- ilość "żyć"
	POKE 14605,100	- nie odejmuje "żyć"
	SYS 14848	
SHARK ATTACK	POKE 6185,100	- nie odejmuje "żyć"
	SYS 8192	
SPLITZ	POKE 5063,0	- zatrzymanie upływu czasu
	POKE 4137,x	- ilość "żyć"

C-16

	POKE 6027,100	- nie odejmuje "żyć"
	SYS 4128	
PUNCHY	POKE 4144,x	- ilość "żyć"
	POKE 4236,100	- nie odejmuje "żyć"
	SYS 4112	
TAPPER	POKE 4240,x	- ilość "żyć"
	POKE 8081,0	- nie odejmuje "żyć"
	SYS 14256	
ZODIAC	POKE 7848,x	- ilość "żyć"
	POKE 8146,0	- nie odejmuje "żyć"
	SYS 7748	
BOULDER DASH	POKE 5961,55	- "FIRE" - przejście do następnej planszy
	POKE 6321,170	- nie odejmuje "żyć"
	SYS 4096 (RUN)	
DIAMOND	- nieśmiertelność uzyskuje się poprzez wprowadzenie HUNTER w linii 1080 instrukcji LI=LI; uruchomienie gry - RUN.	

Kolejną porcję chwytów nadesłał kol. Jarek z Warszawy.

	POKE:	
KIKSTART	10403,197	- "życia" i czas
	SYS 8192	
COOLER	6648,234	- "życia"
	SYS 4352	
BUBBLE	7030,234	- "życia"
TROUBLE	8021,234	- "bańki"
	SYS 12288	
ENIGMA +4	8060,234	- "duchy stop"
	SYS 18432	
SHARK	8210,x	- ilość "żyć" (x=1-255)
ATTACK	6183,173	
	6180,173	- "życia"
	SYS 8192	
TAPPER	20624,x	- ilość "żyć"
	SYS 20481	
GALAXIONS	9635,234	- "życia"
	SYS 4768	
DANGER ZONE	6062,173	- "życia"
	SYS 5816	
AARDVARK	9664,234	- "życia"
	SYS 9458	
CAVE FIGHTER	4922,234	- "życia"
	5004,234	
	5005,234	- czas
	5006,234	
	SYS 8182	
PANCHO	7458,234	
	7459,234	- "życia"
	SYS 4112	
PUNCHY	4234,234	- "życia"
	5057,0	- "czas"
	SYS 4112	
BOOTY	9013,234	
	9014,234	- "życia"
	SYS 4120	
TUTTI FRUTTI	8016,234	- "życia"
	SYS 8192	
CYBORG	10692,234	
	10693,234	- "życia"
	10694,234	
	SYS 11103	
OUT ON ALIMB	5067,234	
OUT ON ALIMB 2	5068,234	
	SYS 5885	
AUTO ZONE	9454,234	
	9455,234	- "życia"
	9456,234	
	SYS 10693	
PLANET	9046,234	
SEARCH	9047,234	- "życia"
	9048,234	
	SYS 12674	
ASTRO	5435,234	- "życia"
PLUMBER	SYS 4100	
GUNLAW	4627,173	- "życia"
	SYS 4120	
JOEY	5760,173	- "życia"
	SYS 4128	
GUNSLINGER	10425,173	- "życia"
	10212,173	
	10317,206	- "broń"
	SYS 8700	
TOWER	12742,160	
OF EVIL	65299,17	
	820,99	- "życia"
	821,20	
	SYS 5120	
VIDEO	10684,173	- "życia"
MEANIES	SYS 10768	
GUZZLER	9025,173	- "życia"
	6452,173	
	4096,162	- "bomby"
	4098,169	
	SYS 13968	
GHOST'S'N'	4365,173	- "życia"
GOBLINS	7783,173	
	65298,251	- "czas"
	SYS 4126	
GNASHER	8946,165	- "życia"
	SYS 5376	
SPACE SWEEP	9077,173	- "życia"
	SYS 8192	
SPECTIPEDE	6791,165	- "życia"
	SYS 4218	
LOCO CO CO	12546,173	- "życia"
	12253,173	- "czas"
	SYS 4867	

OD REDAKCJI. PRZEPRASZAMY AUTORA ARTYKUŁU "NIEZNISZCZALNI" (NR. 4/92) ZA ZNIEKSZTAŁCENIE NAZWISKA, PRAWIDŁOWO BRZMI ONO: ANDRZEJ T. WITBROT.

Otrzymujemy od Was coraz więcej listów z prośbami o pomoc oraz z różnymi pytaniami dotyczącymi Waszych problemów wynikłych podczas pracy z C64. Niestety, mimo najszczerszych chęci nie jesteśmy w stanie odpowiedzieć na każdy list. Dlatego też przedstawiamy kolejną porcję odpowiedzi na kilka wybranych, najczęściej powtarzających się pytań.

1. W różnych publikacjach o C64 spotkałem się z infromacjami o książce "Commodore 64 Reference Guide", która zawiera podobno szersze informacje na temat grafiki oraz możliwości dźwiękowych C64. Gdzie można nabyć taką książkę i ile ona może kosztować?

Istotnie książki takie istnieją, jednak omawiają one w głównej mierze podstawowe możliwości graficzne i dźwiękowe C64. Podane w nich przykłady to w wiekszości dość proste programy w BASICu, które umożliwią szybkie zrozumienie pewnych prawideł rządzących tworzeniem różnych już znacznie bardziej zawansowanych efektów. Książki takie można nabyć w tej chwili już nie tylko na giełdach, gdzie dominują różnego rodzaje kopie tych wydawnictw, ale także w specjalistycznych księgarniach komputerowych czy też sklepach zajmujących się sprzedażą sprzętu komputerowego. Widziałem także w miarę elegancko wydane tłumaczenia tychże książek. Ceny są bardzo różne. Od człowieka pozbywającego się tej literatury można na giełdzie kupić je nawet za cenę około 10.000 złotych. Zwykle jednak ceny zależne są od ilości stron i estetyki wydania i mogą dochodzić nawet do 100tys. złotych.

2. Czy możliwe jest wyświetlenie dwóch lub więcej duszków w tym samym czasie i jak tego dokonać?

Jendnoczesne wyświetlenie nawet 8 duszków nie stanowi większego problemu, gdyż C64 ma możliwość sprzętowej obsługi właśnie takiej ich ilości. Wystarczy wpisać odpowiednie wartości do rejestrów odpowiedzialnych za kolejne duszki. Rejestry te oraz metody korzystania z duszków zostały już dość dokładnie omówione w numerach 3/92 oraz 4/92 naszego pisma. Problem rozpoczyna się dopiero, gdy chcemy wyświetlić więcej niż 8 duszków jednocześnie. Jest to jednak temat na osobny artykuł.

3. Czy można przerwanie rastra umieścić we właściwym programie, a nie w przerwaniu?

Zależy co rozumiemy przez przerwania rastra. Zastosowanie przerwań poza przerwaniem oczywyście nie jest możliwe... Jednak z powodzeniem można użyć w programie właściwym następującej sekwencji:

```
            LDA #$nr_linii
porówanie   CMP $D012
            BNE $porównanie
```

Rozkazy te będą działały bardzo podobnie do zwykłego przerwania rastra. To znaczy program będzie czekał, aż zostanie osiągnięta dana linia ekranu i wtedy dopiero procesor zezwoli na zakończenie wykonywania pętli. Tak więc umieszczenie odpowiednich rozkazów w dalszej części programu i zakończenie ich skokiem do pierwszego rozkazu pętli (LDA #$nr_linii) da taki sam rezultat jakbyśmy używali przerwań. Gdy korzystamy z tej możliwości najwygodniej jest po prostu zabronić wykonywania przerwań rozkazem SEI, dzięki czemu nasz program będzie działał bez żadnych przeszkód.

4. Dlaczego gdy procedurę odtwarzania muzyki umieszczę w programie właściwym i podczas odtwarzania naciskam jakiś klawisz ona zwalnia?

Aby program ten wykonywał się bez żadnych spowolnień należy zlikwidować odwołania do procedury obsługi klawiatury zawartej w ROM'ie. Każde naciśnięcie klawisza powoduje wykonanie serii specjalnych procedur, które interpretują odpowiednio wykonaną przez użytkownika systemu czynność. Zabiera to trochę czasu, więc odgrywana poza przerwaniem muzyka jest trochę opóźniana. Aby temu zapobiec mamy dwie podstawowe możliwości. Albo zrezygnować ze stosowania przerwań i na początek naszego programu wstawić rozkaz SEI albo, jeżli nasz program korzysta również z przerwań, wstawić na koniec przerwania zamiast skoku do obsługi klawiatury (JMP $EA31) skok do bezpośredniego wyjścia z przerwania (JMP $EA7E). Zastosowanie jednej z tych metod pociaga jednak za sobą konieczność czytania klawiatury za pomocą rejestrów CIA $DC00 oraz $DC01.

5. Jak zlikwidować "nierówność" przy wyświetlaniu rastra?

Prawdopodobnie problem polega na tym, iż wyświetlane kolorowe pasy nie mogą się równo ułożyć na ekranie. Otóż, oprócz konieczności zsynchronizowania wyświetlanych pasków przy pomocy komórki $D012, należy je jeszcze odpowiednio ustawić w poziomie. Niestety nie ma żadnego specjalnego rejestru odpowiedzialnego za pozycję wiązki elektronów w poziomie, należy więc skorzystać z odpowied-

C-64

nio dobranych opóźnień. Stosuje się do tego zasadniczo tylko dwa rozkazy, jeden z nich (dwa cykle) to oczywiście NOP oraz drugi, (trzy cykle), to BIT $EA. Warto również zwrócić uwagę na to, iż za pomocą tych rozkazów "docyklowywane" są takie efekty jak wyświetlanie sprite'ów na bocznej ramce, czy różnego rodzaju splity kilkakrotne zmiany kolorów ekranu lub ramki w jednej linii).

6. Gdzie można dostać przystawkę digitalizera do C64?

O ile mi wiadomo jedynym miejscem, gdzie można je spotkać są rożne giełdy komputerowe. Cena tego rodzaju przystawki nie powinna przekroczyć około 300tys. złotych. Program do jej obsługi zwykle jest dołączany do każdego egzemplarza. Jezeli nie to warto spróbować współpracy z zamieszczonym na PDP SampleMon, gdyż jest to jeden z najlepszych tego rodzaju programów dostępnych na C64. Warto jednak wiedzieć, iż jakość uzyskanych za pomocą samplera dźwięków nie jest najlepsza i w żaden sposób nie jest porównywalna z dźwiękami możliwymi do uzyskania na Amidze. Piszę tą uwagę, ponieważ w kilku polskich wydawnictwach ukazały się nieco zawyżone sądy o możliwościach C64.

7. Kilka tygodni temu kupiłem moduł FINAL III do C64. Na początku działał bez problemów, jednak po jakimś czasie zaczął się coraz częściej zawieszać i wyświetlać na ekranie dziwne znaczki. Co jest powodem takiego zachowania?

Zapewne FINAL III - o którym mowa - jest jedną z wielu rodzajów kopii produkowanych przez polskich elektroników. Oprócz niewątpliwej zalety tych modułów, jaką jest ich dość niska cena, mają one jednak także wady. Pierwszą i chyba najaważniejszą jest stosowanie źle dopasowanych do portu pudełek. Powoduje to niepotrzebne i niekontrolowane ruszanie się modułu włączonego do komputera. Taki stan rzeczy w najgorszym przypadku może nawet doprowadzić do trwałego uszkodzenia sprzętu, zwykle jednak dochodzi jedynie do zablokowania systemu i wyświetlenia na ekranie zupełnie bezsensownych znaczków. Często powodem może być również wyrobienie się styków pod wpływem częstego wyjmowania i wkładania modułu. Naprawa tego defektu polega jedynie na odpowiednio delikatnym odgięciu wszystkich styków w porcie. Należy jednak uważać, aby nie wyłamać żadnego z nich, gdyż wtedy zmuszeni będziemy wymienić całe złącze. Trzecia możliwość to zanieczyszczenie styków modułu. Wtedy wystarczy przy pomocy ściereczki i odrobiny denaturatu je wyczyścić.

Ostanią możliwością jest sprzętowe uszkodzenie portu lub modułu. Zwykle w takim przypadku należy zanieść obie rzeczy do serwisu, gdyż sprawdzanie działania modułu u kolegi może się zakończyć naprawą dwóch (zamiast jednego komputerów).

8. Czy można programowo zabezpieczyć dysk przed zapisem?

Można i dość łatwo. Potrzebny jest jedynie monitor dyskowy z możliwością edycji pojedynczych sektorów lub moduł FINAL III czy ACTION REPLAY.

Jak wiadomo dyskietka C64 ma 35 ścieżek (od 1 do 35), każda ścieżka jest podzielona na sektory (w ilości zależnej od położenia ścieżki na dysku). Katalog dyskietki zapisany jest na scieżce numer 18 i zaczyna się od sektora 00, gdzie zapisany jest tzw. BAM dyskietki, czyli tablica zajętości sektorów, nazwa dyskietki oraz jej identyfikator. Od sektora 01 umieszczone są informacje o kolejnych plikach (więcej na ten temat w osobnym artykule). Nas jednak interesować będzie jedynie sektor o numerze 00. Dwa jego pierwsze bajty to numer następnego sektora - zwykle ustawione są na $12, $01. Trzeci bajt tego sektora jest standardowo ustawiony na $41 (czyli literka A), jeśli zmienimy jego wartość na $42 (czyli literka B) i następnie damy rozkaz initialize dla stacji otrzymamy zabezpieczony przed zapisem dysk. Dysk ten nie da się także sformatować bez indeksu. Odbezpieczyć dysk można jedynie przy pomocy specjalnych programów np. Disk Wizard. Programy te dają także możliwość zabezpieczenia w ten sposób dysku bez potrzeby znajomości wszystkich podanych tu informacji, ale wydaje mi się, że warto znać sposób w jaki komputer wykonuje niektóre polecenia.

Jarosław Horodecki

OPIS ZESTAWU PUBLIC DOMAIN PACK

nr 17 (dysk - maj '92)

Mamy przed sobą kolejny Public Domain Pack dla komputerów Commodore 64. Oto opis programów.

Darkside #04 - to kolejny numer najpopularniejszego wśród użytkowników polskiej sceny C64 magazynu. Jak zwykle na początku widzimy nową, zmienianą co miesiąc, czołówkę. Po naciśnięciu spacji i samoczynnym rozpakowaniu się programu wchodzimy do głównej części, w której mamy do wyboru szereg ciekawych artykułów. Autorzy przekazują w nich najnowsze wieści ze scen w innych krajach oraz oczywiście z Polski. Znajdziemy także notowania najlepszych koderów, muzyków, grafików itp. Znajdziemy także dział z adresami osób poszukujących nowych kontaktów. Jest też wywiad, tym razem z koderem oraz liderem polskiej grupy Parados - Rekinem. Autorzy nie zapomnieli także o serii całkiem niezłych dowcipów. Na koniec możemy przeczytać skrótowe sprawozdanie z targów Infosystem'92, które odbyły się w Poznaniu. Magazyn jest całkiem niezły pod względem grafiki, także zastosowana muzyka daje się słuchać. Obsługa magazynu jest bardzo wygodna. Artykuły wybiera się z menu, a następnie czyta przełączając kolejne strony.

Sid'n'Vic #01 to kolejny Polski magazyn dyskowy, tym razem rodem z Wrocławia. Autorzy magazynu wzorowali się częściowo na amigowskim magazynie Fat Agnus. Redakcje tych magazynów także współpracują ze sobą.

Po wczytaniu oraz rozpakowaniu oglądamy elegancko wykonaną planszę tytułową, a po naciśnięciu klawisza space i odczekaniu chwili przechodzimy do głównej części. Wybór artykułów odbywa się także z menu, jednak tym razem to nie podświetlenie się przesuwa, ale cały spis treści (co umożliwi w przyszłości łatwe wstawienie nawet dość dużej ilości artykułów). Warto zwrócić uwagę na to, iż jest to jedyny magazyn na C64, w którym artykuły są przechowywane w pamięci w postaci spakowanej. Umożliwi to z pewnością umieszczanie w kolejnych numerach sporej ilości tekstów. Obsługa magazynu jest dość wygodna i opiera się, podobnie jak w Darkside, na sterowaniu przekładaniem kolejnych stron za pomocą joysticka.

Autorzy magazynu w pierwszym numerze przedstawiają szereg artykułów wstępnych "Od redakcji", czy też "Urodziłem się...". Oprócz tego w numerze możemy znaleźć kilka nowinek mogących zainteresować posiadaczy C64. Warto zwrócić także uwagę na dość interesujący artykuł na temat możliwości rozwoju oprogramowania typu shareware w Polsce. Wszystkich początkujących zainteresują zapewne skrótowe recezje wszystkich najbardziej popularnych modułów dla komputerów Commodore. Autorzy magazynu nie zapomnieli także o rozrywce, np. znaleźć możemy zestaw najpopularniejszych praw Murhy'ego. Podsumowując, muszę stwierdzić, iż jest to jeden z najciekawszych magazynów dyskowych wydawanych w Polsce.



Style #01 to następny nowy magazyn dyskowy dla C64. Tym razem jego autorami są członkowie grupy Skylight. Podobnie jak w innych tak i w tym przed rozpoczęciem czytania tekstów zostaje przedstawiona czołówka grupy wydającej magazyn, w której umieszczone są wszystkie pseudonimy autorów wraz z ich wkładem pracy w pierwszy numer. Przyjemnym urozmaiceniem jest możliwość zmiany muzyki w czołówce.

Po naciśnięciu klawisza space przechodzimy do głównego programu. I tu również grupa Skylight nie zawodzi. Magazyn wykonany jest bardzo estetycznie i pomysłowo. Do wyboru artykułów służy, tak jak w innych tego typu produkcjach, menu zawierające spis treści. Muzyka, z której skorzystali autorzy magazynu jest również dobrze dobrana i nie przeszkadza w czytaniu.

Pod względem tesktu magazyn ten nie wyróżnia się specjalnie spośród produkcji innych polskich grup. Możemy w nim znaleźć wszystkie standardy: zestaw najświeższych wiadomości, notowania najlepszych oraz skrzynkę adresową. Oprócz tego znajdziemy też kącik kodera, a w nim kilka ciekawych i przydatnych procedurek. Możemy też zapoznać się z aktualnym stanem szwedzkiej sceny na C64, czy też przeczytać jak działają moduły typu Final III oraz Action Replay. Na koniec umieszczono relację ze spotkania użytkowników C64 oraz Amigi jakie odbyło się w marcu w Warszawie.

Ogólnie rzecz biorąc magazyn ten jest chyba najstaranniej wykonany ze wszystkich wydawanych dotąd w Polsce. Również poziom artykułów nie jest zły, jednakże autorzy mogliby pokusić się o zastosowanie nieco większej ilości ciekawych pomysłów na nowe teksty.

Shade Music Collection to najnowsza kolekcja muzyczek wydana przez grupę Flash Inc. Są one skomponowane przez muzyka o pseudonimie Shade. Muzyczki te można wykorzystać do pracy z programem Voicetracker V4.0, gdyż są one stworzone przy jego pomocy.

Elysion jest to najnowsze demo grupy Origo. Obecnie jest ono uznane za jedną z najlepszych demonstracji napisanych dotychczas na C64. Autorzy dema sporo wysiłku włóżyli w zaprogramowanie wielu ciekawych i niełatwych do osiągnięcia efektów graficznych. Do najciekawszych z pewnością należą dość sporej wielkości wypełniane wektory, fraktale tworzone w trybie FLI, czy też bardzo ładnie wyglądająca plazma.

Demo to było prezentowane na spotkaniu użytkowników komputerów C64 i Amigi w Szwecji i zdobyło na nim jedno z czołowych miejsc.

Raven's Show I to kolekcja obrazków narysowanych przez Ravena należącego do Wrocławskiej grupy Guns'n'-Roses. Po przejściu przez niezbyt udaną czołówkę możemy zapoznać się z twórczością tego grafika. Obrazki przeglądamy za pomocą klawiszy plus oraz minus. Do każdego z obrazków dołączona jest także muzyczka. Ogólnie demo to nie jest najlepsze, jednakże grafika stoi na całkiem niezłym poziomie.

Do zobaczenia za miesiąc!

Jarosław "Jarri" Horodecki

C-64

W tym roku na łamach naszego pisma publikowaliśmy już artykuły poświęcone grafice na C64. Było to bardzo skrótowe omówienie sposobów wykorzystania sprite'ów z poziomu języka basic. W tym nowym cyklu zajmiemy się grafiką w pełnym tego słowa znaczeniu. Rozpoczniemy oczywiście od prostych przykładów w języku basic, z czasem zajmiemy się jednak także językiem wewnętrznym.

KĄCIK POCZĄTKUJĄCEGO GRAFIKA

Zacznijmy od samego początku. Jak zapewne wszystkim wiadomo nasz komputer ma całkiem niezłe możliwości graficzne. Aby umieć je wykorzystać trzeba być dobrym programistą i znać język maszynowy procesora 6510. Właściwie - moim zdaniem - nie zostały jeszcze odkryte wszystkie sztuczki możliwe do "wyciągnięcia" z VICa. My jednak nie będziemy się zajmować sztuczkami osiągalnymi tylko z poziomu assemblera, ale raczej wykorzystaniem grafiki do przedstawiania różnego rodzaju powierzchni, figur, wykresów. Teraz trochę teorii.

VIC, czyli procesor graficzny C64, posiada dwa podstawowe tryby pracy. Pierwszy z nich to tryb tekstowy, który umożliwia wyświetlenie na ekranie 25 wierszy tekstu w 40 kolumnach. Drugim jest tryb graficzny, który umożliwia wyświetlenie w każdej z 200 linii ekranu 320 punktów. Oba te tryby posiadają swoje odpowiedniki pracujące jako tryb wielokolorowy, którego włączenie powoduje zmniejszenie rozdzielczości poziomej do 160 pikseli dla tworzonego obrazu, ale za to mamy do dyspozycji dwa dodatkowe kolory atramentu. Trybem tekstowym nie będziemy się zajmować, gdyż za wyjątkiem osiągalnych z klawiatury różnych znaczków nie wiele ma on wspólnego z grafiką, natomiast tryb wielokolorowy odkładamy na później. Tak więc pozostaje jedynie hi-res, czyli tryb najwyższej rozdzielczości. Niestety, jak już wcześniej zaznaczyłem, programowanie grafiki na C64 nie jest wcale łatwe i banalne.

Programiści systemu tego komputera nie pomyśleli niestety o jego możliwościach graficznych i z poziomu basic'a nie są dostępne żadne komendy służące do ich wykorzystania. Same dane o grafice w C64 także są ułożone w sposób dość specyficzny, ale o tym za chwilę.

Obrazek w najwyższej rozdzielczości zajmuje w pamięci komputera dokładnie 8192 bajty. Do tej liczby należy dodać jeszcze 1000 bajtów pamięci koloru. Jak więc widać korzystanie z grafiki pociąga za sobą spore straty.

Istnieje osiem możliwych położeń mapy bitowej obrazka, po dwie możliwości w każdym z czterech banków VICa. Aby sobie nieco ułatwić pracę na początek skorzystamy ze standardowo uruchamianego banku znajdującego się od adresu 0 do 16384. Mapa bitowa naszego obrazka będzie umieszczona wtedy od adresu 8192 do 16383, a kolory tak jak standardowa pamięć ekranu tekstowego, czyli od adresu 1024 do 2024. Konfigurację taką ustawi nam następujący zestaw komend:

```
10 POKE 53272, PEEK (53272) OR 8
20 POKE 53265, PEEK (53265) OR 32
```

Linia 10 powoduje ustawienie banku adresu mapy bitów na drugą mapę w aktualnym banku (u nas: 8192), natomiast ekran atrybutów pozostaje bez zmian (adres: 1024). Linia 20 powoduje włączenie trybu graficznego. Proponuje uruchomić ten krótki programik i zapoznać się z jego działaniem. W wyniku jego działania uzyskamy ekran wypełniony najczęściej losowo porozrzucanymi punktami. Możemy także pisać, jednak zamiast liter uzyskujemy tylko różne kolory. Dzieje się tak dlatego, że mimo przejścia w tryb graficzny cały system stale działa, wpisując na ekran nowe znaki zmieniamy zawartość pola atrybutów obrazu, gdyż pamięć atrybutów jest w tym samym miejscu co pamięć ekranu tekstowego. Aby pozbyć się całego bałaganu należy posłużyć się pętlą:

```
30 FOR A=0 TO 999: POKE 1024+A,16*1+0: NEXT A
40 FOR A=0 TO 7999: POKE 8192+A,0: NEXT A
```

Linia 30 to odpowiednie wypełnienie pamięci koloru. Przyjmujemy czarny kolor tła oraz biały kolor atramentu dla całego ekranu. Linia 20 - wypełnienie zerami całego obszaru mapy bitowej obrazka. Jest to odpowiednik instrukcji CLS, czyli czyszczenia ekranu. Jeżeli chcemy korzystać z innych kolorów to wystarczy w linii 30 zmienić jedynkę (biały kolor atramentu) oraz zero (czarny kolor tła) na swoje własne wartości.

Po przygotowaniu ekranu graficznego do pracy musimy zająć się jego wykorzystaniem. Proponuję zacząć od podstawowej czynności, czyli rysowania na ekranie pojedynczego punktu. Zadanie to pozornie łatwe, zostało skutecznie utrudnione przez wprowadzenie przez konstruktorów C64 dość dziwnej konstrukcji mapy bitowej obrazka. Jej ustawienie odpowiada kolejnym znakom o wysokości 8 linijek i szerokości 8 bitów. Tak więc adres lewego górnego rogu ekranu to 8192, ale już następny

bajt w pierwszej linii ekranu to 8192+8, czyli 8200. Natomiast kolejne adresy od 8192 do 8199 to pierwsze bajty kolejnych ośmiu linii ekranu. Z tak skonstruowanych linijek składa się cały ekran C64.

Aby na ekranie postawić punkt należy najpierw obliczyć adres bajtu mapy bitowej obrazu zawierający podaną pozycję, a następnie znaleźć odpowiedni numer bitu. Najpierw obliczymy adres początkowy całego wiersza, w którym mamy postawić punkt. Przyjmijmy iż jego współrzędne to X oraz Y. Aby znaleźć początek odpowiedniego wiersza należy obliczyć ilość wierszy znajdujących się nad naszym punktem. Wykonujemy to następującą instrukcją:

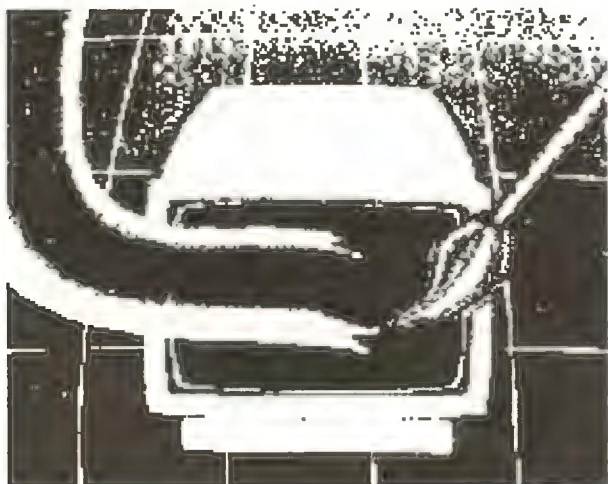
INT(Y/8)

Teraz, aby otrzymać ilość bajtów, jaką należy "przeskoczyć" otrzymany wynik należy pomnożyć przez ilość bajtów w każdym z wierszy. Nasz wzór będzie już wyglądał w następujący sposób:

INT(Y/8)*320

Teraz musimy tylko wiedzieć, który bajt w danym wierszu należy zapalić. Obliczymy oczywiście ilość znaków poprzedzających nasz bajt a następnie, aby otrzymać tę informację w bajtach, pomnożymy ją przez 8.

INT(Y/8)*320+INT(X/8)*8



Ostatnią czynnością będzie dodanie do naszych obliczeń odpowiedniego przesunięcia, aby uzyskać szukany bajt w znalezionym już znaku. W tym celu obliczyć musimy resztę z dzielenia X/8. Na końcu dodajemy jeszcze początek naszej mapy bitowej i otrzymujemy obliczony adres naszego punktu:

ADRES=8192+INT(Y/8)*320+INT(X/8)*8+(Y-8*INT(Y/8))

Teraz wystarczy tylko znaleźć numer bitu, który należy zapalić. Przy czym, aby nie modyfikować zawartości pozostałych bitów w znalezionym bajcie, posłużymy się funkcją logiczną OR. Numer bitu, który należy zapalić znajdujemy obliczając resztę z dzielenie X/8. A konkretną liczbę, jaką musimy podstawić do funkcji OR otrzymujemy po podniesieniu tej dwójki do potęgi tej liczby.

Oto gotowy podprogram postawienia punktu na ekranie:

```
999 END
1000 ADRES=8192+INT(Y/8)*320+INT(X/8)*8+(Y-INT(Y/8)*8)
1010 POKE ADRES, PEEK (ADRES) OR 2^(7-(X-INT(X/8)*8))
1020 RETURN
```

Linia 1000 to oczywiście wyliczenie adresu puntu w mapie bitowej, linia 1010 korzystając z wcześniejszych obliczeń stawia na ekranie punkt. Linia 1020 to powrót z podprogramu. Linia 999 została wprowadzona tylko dlatego, aby program główny nie wchodził w obszar podprogramów powodując błąd i zatrzymanie realizacji programu.

Mając gotową procedurę wyświetlania punktu warto pokusić się o napisanie programów wykorzystujących ją. Pierwszy z nich będzie rysował na ekranie 1000 punktów o losowo wybranych współrzędnych. Korzystamy oczywiście z wpisanych dotąd linii. Na koniec programu dopiszmy jeszcze następującą sekwencję:

```
990 GET A$: IF A$<>" " THEN 990
991 POKE 53272, PEEK (53272) AND NOT 8
992 POKE 53265, PEEK (53265) AND NOT 32
993 PRINT CHR$(147);
```

Sekwencja ta spowoduje oczekiwanie programu na naciśnięcie klawisza spacji, następnie wyłączy tryb graficzny oraz skasuje zawartość ekranu tekstowego. Teraz możemy już przejść do pisania naszego pierwszego programiku:

```
100 FOR A=0 TO 999 110 X=INT(RND(0)*319)
120 Y=INT(RND(0)*199)
130 GO SUB 1000
140 NEXT A
```

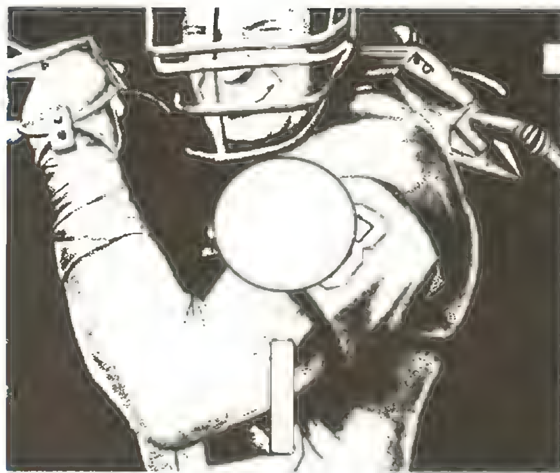
Program wykonuje losowanie współrzędnych, a następnie skacze do pdprogramu rysowania punktu na ekranie. Czynność ta wykonywana jest 1000 razy. Prawda, że wygląda całkiem ładnie?

Jako zadanie domowe proponuję napisać procedurę kasującą zadany punkt oraz wzbogacić nasz przykładowy programik o efekt migania losowo wybranych gwiazdek (tylko już tych wyświetlonych!).

Jarosław "Jarri" Horodecki

C-64

DYSK



Chyba każdy wie co to jest dyskietka i do czego służy. Nie każdy jednak orientuje się jak zapisane są na niej dane i co można w nich zmieniać, aby uzyskać ciekawe efekty w jej katalogu.

Na C64 napisano już bardzo wiele programów służących do robienia różnych zmian w katalogach. Dzięki nim możemy tworzyć "skasowane" pliki, które w katalogu są wyświetlane jako pliki typu DEL, możemy też dowolnemu plikowi nadać fikcyjną długość począwszy od 0 a skończywszy na - niemożliwej do zapisania na dysku ilości - 65535 bloków, możliwe jest także dowolne zmienianie typów każdego z plików, czy też ich zabezpieczanie lub odbezpieczanie. Wszystkie te możliwości zwykle dostępne są w lepszych edytorach katalogu dysku, z pewnością jednak warto znać sposób w jaki wszystkie te sztuczki są realizowane. Zacznijmy od podstawowych informacji.

Każda dyskietka po sformatowaniu podzielona jest na pewną liczbę ścieżek, a każda ze ścieżek podzielona jest na pewną ilość sektorów. Większość komputerów bez względu na położenie ścieżki nadaje każdej z nich tą samą liczbę sektorów. Jednakże twórcy stacji dysków do C64 oprócz „niesamowitej" prędkości transmisji wprowadzili także różne ilości sektorów na ścieżkach. Ilość ta zależy od położenia danej ścieżki względem środka nośnika. Dysk sformatowany w stacji do C64 podzielony jest na 35 ścieżek o numerach od 1 do 35, które dzielą się w następujący sposób: ścieżki od 1 do 17 mają po 21 sektorów, od 18 do 24 po 19 sektorów, od 20 do 30 po 18, od 31 do 35 po 17 sektorów. W każdym z sektorów można zapisać 256 bajtów danych. Łatwo więc wyliczyć, ile w sumie możemy zapisać bajtów informacji na jednej strone dyskietki. Wyliczona w ten spsób pojemność nie jest jednak w pełni prawdziwa, gdyż należy jeszcze odliczyć sektory zajmowane przez katalog oraz specjalne bajty informujące o następnym bloku, ale o tym nieco później.

Znamy już zasadniczą architekturę naszego nośnika. Teraz kilka słów o tym, jak właściwie zapisane są na nim dane. Otóż najważniejszą rzeczą jest katalog dyskietki czyli spis wszystkich zapisanych informacji. Umieszczony on jest na całej 18 ścieżce. Jeden z sektorów tej ścieżki przeznaczony jest na opis zajętości bloków całej dyskietki oraz jej nazwę i identyfikator, natomiast pozostałe sektory przeznaczone są tylko na informacje o kolejnych plikach. Jak łatwo zauważyć w sumie na te informacje pozostaje 18 sektorów, czyli dokładnie 4608 bajtów. Dane o każdym z plików zajmują 32 bajty, więc na dyskietce możemy zapisać maksymalnie 144 pliki.

Propunuję teraz zająć się nieco dokładniej właśnie ścieżką 18, gdyż znajdują się na niej wszystkie potrzebne nam do wykonania różnych sztuczek z katalogiem informacje. Do naszej zabawy potrzebny będzie dowolny monitor dyskowy z możliwością zmieniania zawartości dowolnego bajtu w dowolnym sektorze lub też monitor języka maszynowego posiadający możliwość odczytywania i zapisywania sektorów. Doskonale do tego nadają się monitory umieszczone w modułach Final III oraz Action Replay, przy czym drugi z nich jest nieco wygodniejszy w obsłudze.

Co się właściwie dzieje, gdy wczytujemy katalog dyskietki? Otóż komputer przy pomocy specjalnych procedur umieszczonych w pamęci ROM odczytuje kolejne sektory ścieżki 18 począwszy od sektora 00. Po jego odczytaniu na ekranie pojawia się nazwa oraz identyfikator dyskietki. Następnie komputer przechodzi do kolejnego sektora. Skąd jednak ma wiedzieć jaki właściwie sektor jest tym następnym? Proponuję spróbować wczytać do pamięci sektor 00 ścieżki 18.

W module Action Replay posłuży do tego następująca komenda:

@BR 12 00 20

(Wszystkie wartości podane są w systemie szesnastkowym, gdyż w takim tylko systemie pracują praktycznie wszystkie monitory języka maszynowego na C64.) Komenda ta spowoduje wczytanie podanego sektora do pomięci pod adres - w naszym przypadku - $2000. Należy zwrócić uwagę na to, aby w tym samym miejscu pamięci nie znajdowały się istotne dla nas dane, gdyż wczytanie w ich miejsce zawartości sektora spowoduje ich nieodwracalne skasowanie. Teraz przy pomocy komendy:

I*2000

możemy zapoznać się z danymi umieszczonymi w tymże sektorze.

Na początku umieszczone są jakieś trzy bajty, potem troche dziwnych znaczków i w końcu nazwa oraz identyfikator naszej dyskietki. Dwa pierwsze bajty w tym (i także każdym innym) sektorze to właśnie kolejno: numer ścieżki oraz numer sektora, które mają zostać odczytane w następnej kolejności. Dalsze bajty to już zasadnicza zawartość danego sektora. W tym miejscu możemy wprowadzić poprawke do naszych obliczeń pojemności dyskietki, gdyż każdy sektor mieści jedynie 254 bajty "czystej" informacji. Ale wróćmy do naszego zerowego sektora ścieżki katalogu.

Otoż trzeci bajt w tym sektorze to informacja o programowym zabezpieczeniu dyskietki przed zapisem jak również przed przypadkowym sformatowaniem bez indeksu. Normalnie znajduje się tam wartość $41 (duża litera A), aby jednak dysk został zabezpieczony należy zmienić ją na duża literkę B, czyli kod $42. Po zapisaniu sektora z pamieci na dyskiekę (komendą @BW12 00 20) oraz zainicjowaniu stacji (komendą @I - initialize) spróbujmy zapisać lub skasować dowolnie wybrany plik. Komputer zawsze będzie wyświetlał informację, iż dysk jest zabezpieczony. To samo nastąpi także podczas próby formatowania dyskietki z podaniem samej nazwy, bez indeksu.

Zajmijmy się jednak dalszą zawartością tegoż sektora. Seria dziwnych znaczków jakie następują po tych trzech bajtach to tablica zajętości sektorów dyskietki (tzw. BAM). Komputer korzysta z niej przy zapisie danych, aby sprawdzić czy dany sektor jest jeszcze wolny, czy też znajduje się już na nim wcześniej zapisana informacja. Tablica ta jest także modyfikowana podczas kasowania pliku z dyskietki. Zmiany w niej dokonywane mogą doprowadzić do uzyskania różnych dość dziwnych efektów, jak na przykład uzyskanie 1992 bloków wolnych. Gdy na dysku, który w ten sposób przerobiliśmy mamy zapisane ważne informacje polecałbym zabezpieczenie go przed zapisem lub też wykonanie instrukcji @V (validate), która spowoduje odtworzenie starej zawartości BAMu. Więcej o kontrolowanym zmienianiu ilości bloków napiszę za miesiąc. Teraz przejdźmy do dalszej części katalogu.

Spójrzmy na dwa pierwsze bajty sektora 00 (przy pomocy komendy monitora M2000). Zawierają one zwykle wartości: 12 01, co wskazuje na pierwszy sektor osiemnastej ścieżki. Wczytujemy więc podany sektor komendą

@BR12 01 20

i - o ile na dysku zapisane są jakieś pliki - możemy teraz zmieniać w dowolny sposób wygląd katalogu. Oczywiście nie zwracamy uwagi na dwa pierwsze bajty sektora, gdyż jest to po prostu numer kolejnego sektora lub kombinacja 00 FF - jeżeli jest to już ostatni sektor katalogu. Skoncentrujemy się na kolejnych bajtach (będę podawał numer bajtu w każdej linijce sektora (32 bajty) - bajty o numerach 0 oraz 1 są nie używane, a jedynie w pierwszej linijce zawierają numer kolejnego sektora. Bajt numer 2 określa typ danego pliku (PRG, DEL , SEQ, REL lub USR). Jest jeszcze szereg innych typów, które wynikają z błędnego odczytu danych z ROM'u stacji. Podając losowe wartości jako typ pliku otrzymujemy różne bezsensowne kombinacje znaków ASCII. Oto kody wszystkich podstawowych typów plików:

PRG - $82,
DEL - $80,
REL - $00,
SEQ - $01,
USR - $03.

Przez zmienianie starszych czterech bitów możemy także włączać oraz wyłączać zabezpieczenia danego pliku przed zapisem oraz kasowaniem. Jak widać wartość dla zwykłego pliku to $8x , natomiast plik zabezpieczony musi mieć na pierwszej pozycji $C. W katalogu po tej zmianie pojawia się za typem danego pliku znak mniejszości. Można także oznaczyć plik jako niezamknięty. W katalogu pojawia się gwiazdka (znak mnożenia) umieszczona po lewej stronie typu pliku. Oznaczenie to uzyskujemy przez wprowadzenie $0 na pierwszej pozycji bajtu typu pliku. Można oczywiście włączyć obydwa oznaczenia na raz, tzn. zarówno gwiazdkę, jak i znak mniejszości. Uzyskujemy to przez wpisanie $4 jako starszej połowy tego bajtu.

Kolejne dwa bajty to numer ścieżki oraz, jako następny, numer sektora od którego zaczyna się dany plik. Następnie, po jednym bajcie przerwy następuje nazwa odpowiedniego pliku zakończona kodami $A0. Komputer podczas wyświetlania katalogu szuka właśnie kodu $A0 i po znalezieniu go drukuje na ekranie znak cudzysłowia, a następnie odczytuje nazwę pliku. Tak więc umieszczenie w nazwie znaku $A0 oraz dopisanie dalszej jej części spowoduje wyświetlenie tejże części pliku już za cudzysłowiem. Dalej następuje 9 bajtów przerwy, po których podana jest ilość zajmowanych przez dany plik bloków. I tutaj mamy pełne pole do popisu, gdyż długość pliku określona jest przez dwa bajty. Tak więc jak już wcześniej wspomniałem możliwa jest jej zmiana począwszy od 0 do 65535 sektorów. Oczywiście jak zwykle proponuje poeksperymentować z dobieraniem różnych wartości.

I to są właściwie wszystkie podstawowe informacje o samym katalogu dyskietki. Za miesiąc więcej o BAMie oraz o sposobie zapisu plików, a także kilka przykładów wykorzystania poznanych wiadomości we własnych programach w języku Basic oraz assemblerze.

Jarosław "Jarri" Horodecki

C-64

MAPA PAMIĘCI

Na życzenie naszych czytelników publikujemy pierwszą część mapy pamięci C-64.

STRONA ZEROWA: $000 DO $00FF

ADRES HEX	DEC	OPIS
$00	0	rejestr kierunku danych
$01	1	port wejścia/wyjścia
$02	2	nie używane
$03/$04	3/4	wektor procedury przekształcenia liczby zmiennoprzecinkowej na całkowitą ze znakiem (zwykle $b1aa)
$05/$06	5/6	wektor procedury przekształcenia liczby całkowitej na zmiennoprzecinkową (zwykle $b391)
$07	7	używana przez interpreter Basic'a
$08	8	używana przez Basic
$09	9	pozycja kolumny kursora przed otatnim rozkazemTAB lub rozkazem SPC
$0a	10	flaga LOAD(0) lub VERIFY(1)
$0b	11	wskaźnik buforu wprowadzenia, również liczba zmiennych tablicowych
$0c	12	używana przy interpretacji tablic
$0d	13	flaga typu danych (0=liczba, $ff=łańcuch)
$0e	14	flaga danej numerycznej (0=zmiennoprzecinkowa, $80=całkowita)
$0f	15	flaga LIST, przekształceń tekstu i GARBAGE COLECTION
$10	16	flaga pola zmiennych lub zdefiniowanych przez użytkownika funkcji
$11	17	flaga INPUT ($00), GET ($40) lub READ ($98)
$12	18	znak wyniku funkcji SIN, COS, TAN
$13	19	flaga aktualnego kanału wejścia/wyjścia - meldunki ekranowe będą wstrzymywane jeśli różna od 0
$14/$15	20/21	numer linii dla LIST, GOTO, GOSUB, ON. Wskaźnik adresu przy PEEK, POKE, WAIT, SYS
$16	22	indeks następnego wpisania w łańcuch Descriptor Stack
$17/$18	23/24	wskaźnika aktualnego wpisania w łańcuch Desriptor Stack
$19-$21	25-33	tymczasowy łańcuch Descriptor Stack
$22-$25	34-37	różne pamięci chwilowe
$26-$2a	38-42	pamięć robocza arytmetyki
$2b/$2c	43/44	wskaźnik początku programu w Basic'u (zwykle 2049)
$2d/$2e	45/46	wskaźnik początku obszaru zmiennych
$2f/$30	47/48	wskaźnik początku obszaru pola zmiennych tablicowych
$31/$32	49/50	wskaźnik dolnej granicy zakresów łańcuchów
$33/$34	51/52	wskaźnik aktualnej granicy obszaru łańcuchów
$35/$36	53/54	wskaźnik adresu ostatnio wprowadzonego łańcucha
$37/$38	55/56	wskaźnik ostatniej komórki obszaru Basic'a (zwykle 40960)
$39/$3a	57/58	numer aktualnej linii Basic'a
$3b/$3c	59/60	numer linii ostatniego przerwania programu
$3d/$3e	61/62	wskaźnik początku aktualnej linii w Basic'u dla CONT
$3f/$40	63/64	numer linii aktualnej instrukcji DATA
$41/$42	65/66	wskaźnik ostatnio odczytanego bajtu przez READ z instrukcji DATA
$43/$44	67/68	wskaźnik źródła INPUT, GET, READ
$45/46	69/70	nazwa aktualnej zmiennej
$47/48	71/72	wskaźnik adresu wartości aktualnej zmiennej
$49/$4a	73/74	pamięć chwilowa dla różnych rozkazów Basic'a
$4b/$4c	75/76	pamięć chwilowa dla operacji matematycznych
$4d	77	pamięć pomocna dla operacji porównywania
$4e/$4f	78/79	wskaźnik funkcji zdefiniowanych przez użytkownika
$50-$52	80-82	tymczasowy wskaźnik dla adresów i długości łańcuchów
$53	83	długość zmiennych łańcuchowych podczas Garbage Collection
$54-$56	84-86	rozkaz skoku do adresu aktualnej funkcji Basic'a
$57-$60	87-96	pamięć robocza procedur arytmetycznych
$61-$66	97-102	akumulator zmiennoprzecinkowy #1(FAC1)
$67	103	pamięć chwilowa i rejestr liczb

PUBLIC DOMAIN PACK

PUBLIC DOMAIN PACK C-64

Styczeń '91 (nr 1)

- Mega demo grupy „VISION"-MIST2, Preview do gier: UN SQUADRON, PUZZLENOID, TURRICAN.

Luty '91 (nr 2)

- Tune of month, Logo Writer V 2.0, Fast Cruelcrunch, WRATH+ (DEMO)[02], Dreptacz - BASIC, SWISS CHEESE/CFA, Disk Fast Loader.

Marzec '91 (nr 3)

- Font Grub 1.0, Projektant Duszków, Strzałka 64+, Piratek - gra, V4.0 - Symphonies, Cruiser, The First, Commercial Break, Relakator 64, Korektor 64, Flash, HOT SHOT nr 9 (zach. mag.), BAD NEWS nr 2 - j.w., demo - rekord - 290 sprite'ów!, demo: NEW INTRO, demo: LET'S DYCP, Kontakt Corner - adresy, New Fast - działa z 1541 I 1541 II, CSLINKER V2.0.

Kwiecień '91 (nr 4)

- Digi - Organizer - program do tworzenia muzyki z użyciem digitalizacji dźwięku, „ONE YEAR - RADIUS" - mega demo grupy RADIUS.

Maj '91 (nr 5)

- CRUEL SOLIDERS - demo, DESTINATION - demo, SUCKER DJ! - demo (digi mix), MUSIC SEARCHER - do wycinania ilustracji muzycznych z programów, MEGA DEMO „INFOSYSTEM 91".

Czerwiec '91 (nr 6)

- Fonteditor, Sindata Editor, Color Editor, Disk - Noter, Gwiazdy - demo graficzne, FILGRAEPH 2.2/BML, NOTE TO FLI V 2.2, AFLI - EDITOR V 1.2, RESET - MON,8,1, TURBO - ASS 5, HIGHLIFE #5, AXEL NEWS #1, DISK NOTKA/PADUA, PSC - MAG #9'06/91, CONSPIRE? OREGON - demo, CONTACT DEMO/ORE, SHOWPIX.

Lipiec '91 (nr 7)

- Mega demo „MY, OH MY!" grupy LIGHT, Game Music Composer - edytor muzyczny grupy GRAFFITY z Węgier.

Sierpień '91 (nr 8)

- MegaDemo „Unnamed" grupy CAMELOT, Sound Killer - edytor muzyczny grupy TOPAZ, AFLI - edytor graficzny techniki A-FLI, Disk-Dos obsługa komend stacji dysków, Noter v2.2 grupy TOPAZ, IFFL - Squeezer kompresor dyskowy, Dismaster+ - edytor do dyskietek, Super Copy - DOS szybki program kopiujący do zbiorów, Mega Demo fińskiej grupy TOPAZ - „Graveyard Blues".

Wrzesień '91 (nr 9)

- Mega Demo grupy FLASH, Hot Shot - magazyn dyskowy, Code Sucker monitor - pr. użytkowy grupy PADUA, Mountain Ride - gra w BASIC.

Październik '91 (nr 10)

- MEGA DEMO „AIRDANCE 4" grupy T.A.T.

Listopad '91 (nr 11)

- NEW LAW & ORDER, FLT/LEGOLAND, FLT/LEGONOTE, TERMINAT .2%/FLT, SM. CRIMINAL #8, SMALL BUT FINE, HOLLY SMOKE/M12, UNITEI/SYLVIO.

Grudzień '91 (nr12)

- Armageddon 3, NOTE TO DEMO, OUTRUN 2 MUS $ SFX, AFTERBURNER/MON, TRIVIA-GAME MUSIC, FORM.I. SIMULATOR, 2400AD END-TUNE, NIGHTHUNTER MUSIC, TOMCAT MUS./MON, ZAMZARA TUNE/MON, NOTE TO DISK, HIGHLIFE #9.

Styczeń '92 (nr 13)

- Char Zoomer v3.1, Colour Bar Editor v3.0, Hires+A-FLI Editor, FLI Designer v1.1, Accesinus, Music Routine Cruncher v1.5, Gandalf Protector v3.0, Gandalf Coder v1.0, Disk-tape copy v.20, Gnd - packer v1.0.

Luty '92 (nr 14)

- LYNX XVI+, Sideborder Logo Editor V1.0, Intelpaint, Flieditor V3.2, 4*4 Charmaker, F(R)ONT EDITOR 3, THE GRAFIX PACK II, DEMA.

Marzec '92 (nr 15)

- Contact Dealer v3.0, Beeftrucker v1, Cross linker v3, PowerCruncer v7.1, Sample mon v.2.0, Handy Term v8.4, Highlife, Darkside, Humor Basic, dema.

Kwiecień '92 (nr 16)

- CIABACH 4&5, DARKSIDE, Convert Studio, edytor znaków grupy Skylight, Sprite Designer, Multicolour Conventer v1.3, Logo Flipper, Charset Maker v1.0.

PUBLIC DOMAIN PACK AMIGA

Styczeń '91 (nr 1)

- Programy kompresorów danych, Grafiki Borysa Vallejo, prezentacja najlepszych muzyczek, INTUITRACKER

Luty '91 (nr 2)

- Request player; Multi ripper, 3-rd day; Phantasmagoria - demo, Master Seka; Virus Ekspert v1.6, AMOS-programy; Moduly: Killing game show, Upon Me, Let's swing it.

Marzec '91 (nr 3)

- PROTRACKER V1.0 (pakiet programowy), Najlepsze muzyczki: NOW WAIT? - DR.AWESOME, AMOS - procedury, DEMO grupy REBELES „TOTAL TRIPLE TROUBLE".

Kwiecień '91 (nr 4)

- RUBBER VECTORS - demo, KEF-TALES - demo, DISK MASTER V3.0, Moduły muzyczne: TECHNOSTYLE 2, GALAXY 2, GRAFIKA - prezentujemy rysunki - RICK PARKS.

Maj '91 (nr 5)

- VIRUS X 5.0, VIRUS TERMINATOR, PARADOX - demo, STORMCHILD - demo, Moduły muzyczne: MIAMI VOICE, ANTI ATARI SONG.

Czerwiec '91 (nr 6)

- POWER BOOT - własne menu dysku, DISK CODING SYSTEM - program do zabezpieczania dysków, Konwerter IFF - ANSI, AUER NATION - demo, Moduły muzyczne, DOCS - opis gry ELWIRA, LAMER DEFENCE - do wykrywania i niszczenia wirusów, REWENG GO OF THE LAMER - grafika w trybie D-HAM.

Lipiec '91 (nr 7)

- Sanity - demo, Amiga - Tanx (1Mb) - gra, Little Beau (1MB) - gra, There is A Light/Tonid - modules.

Sierpień '91 (nr 8)

- Real 3D - demo programu do raytraceing'u, moduł muzyczny XTC STEREO.

Wrzesień '91 (nr 9)

- Moduły muzyczne dla programu TFMX: R - TYPE; The House of Techno;VIRUS EXPERT v181 + 143; Boot Block'i, Bootx v 3.80, Imploder v 4.0.

Październik '91 (nr 10)

- ANARCHY - „THE INSPIRATION IS NONE"; ‚DUAL CREW - „NEW DIMENSION"; SANITY - „ELYSIUM".

Listopad '91 (nr 11)

- COMPUTER HEAD - animacja; CONFUSED - moduły pod medplayer i wiele sampli; ROCKED -; SAVE GAME „MONKEY ISLAND".

Grudzień '91 (nr 12)

- GEM X; BOOTX V4.13; FINAL KIT - monitor; MEGA-MON; VARIA.

Styczeń '92 (nr13)

- Super Duper 2.01; Sanity Copy; Noise Packer 3.00; Ham Sharp; Mostra; dema graf. i muz.

Luty '92 (nr 14)

- Nuke Saddam 1.4; THIEF RIPPER 2.0; DISK MASTER 3.05; ZIG ZAG #3 (grafika); dema graf. i muz.

Marzec '92 (nr 15)

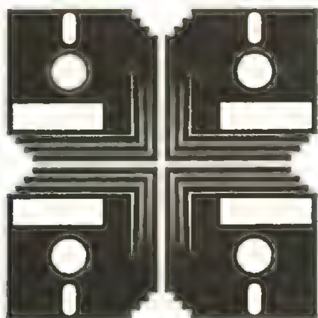
- Gry: KIM, Power Wars; PrtDrvGen; FileMaster v1.1; FixDisk!; Facc; Intro Well!; digitalizacje (IFF).

Kwiecień '92 (nr 16)

- Disk Cruncher; IBM-ST-AMIGA; HamLab v0.91; BAD v4.13; BOOTX v4.47,; DragonTiles - gra; grafiki IFF.

PUBLIC DOMAIN PACK





PUBLIC DOMAIN PACK C-64
TAPE NR 1

- TURBO
- SINUSDATA - EDITOR
- FAST CRUNCHER V3
- ANAL S.C. IBEYOND
- VECTOR - VICTORY
- PUZZLENOID+4
- TUNE OF MONTH #1
- NIM
- STRZAŁKA 64+
- LOGO - WRITER V.2.0
- CAN'T TOUCH IKU!
- NTRO PRV
- BONZIEED!!
- ZAX PACKIS
- READ THIS FIRST
- COMMERCIAL BREAK
- 290 SPRITES!
- NOTE - ABOUT
- BAD NEWS NR2
- TO BAD NEWS...
- CONTACT CORNER!
- PROJEKT DUSZKÓW
- SYMPHONY NR14
- SYMPHONY NR15
- SYMPHONY NR16
- SYMPHONY NR17
- SYMPHONY NR18
- SYMPHONY NR19
- CRUISER/GIANTS
- NOTE>ANO<PADUA
- LET'S DYSP!
- FINALTAPE
- MUSIC - SEARCHER

PUBLIC DOMAIN PACK C-64
TAPE NR 2

- TURBO
- PUBL. DOMAIN. INFO
- FONTGRUB 1.0
- DREPTACZ BASIC
- LOAD DIS FIRSY
- MACROASSEMBLER
- TURBOASSEMBLER
- RELOCATOR
- LOGOPAINTER 3!
- REASSEBLER
- SPRITE - EDITOR
- FAST - CRUEL U.2.5
- HIGHLIFE NR5
- AXEL NEWS NR1
- GWIAZDY
- FLIGRAPH 2.2/BML
- NOTE TO FLI V.2.2
- DISKNOTKA/PADUA
- MEGA PACKER/T
- MIST II/ VISION
- TTECHSCR &DYSP
- PLASMA - WORLD
- VECTORBOBS...
- VECTOR - PLOTS
- FLI - UPSCROLL
- BORDER - HIRES
- ROCK AROUND
- FACEWRITER
- CHAR EDIT 2+2
- DISKNOTER
- DESTINATION'91
- CONTACTDEMO/ORE
- FONTEDITOR
- THE END

PUBLIC DOMAIN PACK C-64
TAPE NR 3

- TURBO
- PUBLIC DOMAIN NOTE
- GRAVEYARD NOTES!
- NOTE FROM BEAT!!
- ANONYM SPEAKING!
- SNDK. V3.7/TOPAZ
- AFLI - EDITOR
- NOTER V2.2/TOPAZ
- DLW V1.5/TOPAZ
- CODE - S.MON/PADUA
- OPINION - POLL/PDA
- MOUNTAIN RAID
- PART 1
- PART 2
- PART 3
- PART 4
- PART 5
- FAIRLIGHT 1
- FAIRLIGHT 2
- FAIRLIGHT 3
- FAIRLIGHT 4
- FAIRLIGHT 5
- THE END

PUBLIC DOMAIN PACK C-64
TAPE NR 4

- TURBO
- OUT RUN 2 MUS & SFX
- AFTER BURNER/MON
- FORM.1.SIMULATOR
- 2400 AD.END - TUNE
- NIGHT HUNTER DIGI
- ELEMINATOR MUSIC
- TOMCAT MUSIX/MON
- ZAMZARA TUNE
- DYNAMIX TUNE
- HIGHLIFE NR9
- SNAKES C3
- SNARK C3
- SNERD C3
- WAREHOUSE C3
- STARTREK C3
- TOWER
- SNOOPY
- NEW LAW & ORDER
- FLT/LEGONOTE...
- TERMINAT.2%/FLT
- UNITE!/SYLVIO
- BALL - SCOPE/451
- TRIVIA - GAME MUS.
- RESET - MONITOR
- HOLY SMOKE

Zestawy „64 plus 4 PUBLIC DOMAIN PACK" można zamawiać wpłacając na konto: Bank PKO SA Oddział w Bydgoszczy konto nr: 5.09011-400522.7-2511-30-111.0 następujące kwoty: 20.000zł za pojedyńczy zestaw dyskowy dla C-64, 30.000 zł za zestaw programów PD na kasecie, 25.000zł za zestaw dla Amigi.

Blankiety wpłat powinny być CZYTELNIE wypełnione i zawierać: **imię i nazwisko, dokładny adres zamawiającego, skrót „PDP-64D"** - jeśli zamawiamy zestaw dla C-64 na dyskietce lub **„PDP-64T"** - dla zestawu taśmowego, zestaw dla Amigi prosimy zaznaczać skrótem **„PDP-A"** - dane te prosimy umieszczać na **wszystkich** odcinkach dowodu wpłaty.

W prenumeracie zestawy kosztują: **PDP-64 - 18.000zł** (12 numerów 216 tys. zł), **PDP-A - 22.000 zl** (12 numerów 264 tys. zł). Prenumeratę można zawrzeć w dowolnym terminie na okres od 3 do 12 miesięcy (do końca roku kalendarzowego). Powyższe warunki odnoszą się również do naszych zestawów wydanych w 1991r .

Zestawy taśmowe PDP-64 w 1992r. będą ukazywały się w miarę napływu nowych, ciekawych programów - o czym będziemy informować na łamach naszego pisma.

Zamów nie zwlekaj!

VOICETRACKER V4.0

C-64

Rewelacyjny program muzyczny!



Tylko **50.000 zł** kosztuje fantastyczny edytor muzyczny wykorzystujący ogromne możliwości dźwiękowe komputera Commodore - 64. Oferowany zestaw zawiera dyskietkę lub taśmę magnetofonową z programem VOICETRACKER V4.0, instrukcję obsługi, oraz - dodatkowo - przykładowe demonstracje muzyczne. **UWAGA! Wersja magnetofonowa tylko 40.000 zł!**

Przedsiębiorstwo ABUK posiada wyłączność na dystrybucję tego programu. Wszelkie kopiowanie programu i powielanie instrukcji jest zabronione. Nabywcy otrzymują rejestrowane kopie programu wraz z prawem nabywania nowych wersji po znacznie obniżonych cenach oraz wymiany dyskietki w razie uszkodzenia. Studiom komputerowym proponujemy zakup hurtowy (przy zakupie powyżej 10 kompletów udzielamy 20% rabatu).
Chcąc stać się posiadaczem programu VOICETRACKER V4.0 wystarczy dokonać wpłaty 50.000 zł (wersja dyskowa) lub 40.000 zł (taśma) na konto: Bank PKO SA Bydgoszcz, konto nr: 5.09011-400522.7-2511-30-111.0. Na blankiecie prosimy czytelnie podać swoje imię, nazwisko i adres wraz z dopiskiem „VV4.0" uzupełnionym literką „T" - taśma lub „D" - dyskietka.

W związku z pojawiającymi się kłopotami w dystrybucji oferowanych przez nas dyskietek i taśm (wynikającymi z nieczytelnego bądź niekompletnego wypełnienia blankietów wpłat) przedstawiamy obok specjalny druk. Blankiet ten może służyć jako zamówienie i dowód wpłaty dla wszystkich oferowanych przez nas usług: sprzdaż dyskietek i taśm PDP, Voicetracker'a, zamówienie ogłoszeń itd. Informujemy, że urzędy pocztowe zobowiązane są do przesłania dwóch odcinków (dla banku i dla wpłacającego) do naszego banku.

REDAKCJA

Odcinek dla wpłacającego

Zł ....
słownie ....
....
wpłacający ....
....
....
(dokładny i CZYTELNY adres)

na rachunek:
Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-2511-30-111.0.

Opłata
zł..........

Odcinek dla posiadacza rachunku

Zł ....
słownie ....
....
wpłacający ....
....
....
(dokładny i CZYTELNY adres)

na rachunek:
Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-2511-30-111.0.

Opłata
zł..........

Odcinek dla Poczty

Zł ....
słownie ....
....
wpłacający ....
....
....
(dokładny i CZYTELNY adres)

na rachunek:
Przedsiębiorstwa ABUK sp. z o.o.
87-200 Wąbrzeźno, ul. 1 Maja 33,
Bank PKO SA Bydgoszcz, konto:
5.09011-400522.7-2511-30-111.0

Opłata
zł..........



TREŚĆ ZAMÓWIENIA:

TREŚĆ ZAMÓWIENIA:

Prosimy o CZYTELNE wypełnienie.

TREŚĆ ZAMÓWIENIA:

Prosimy o CZYTELNE wypełnienie.

Zapraszamy wszystkich do udziału w stałym konkursie pod hasłem:

Najlepszy program miesiąca

W konkursie udział mogą brać wszyscy, którzy nadeślą własne, nigdzie nie publikowane prace. Tematyka programów dowolna. Konkurs rozgrywany jest osobno dla komputerów C-64 i Amiga.
Teksty programów należy nadsyłać na adres redakcji na dyskietce lub w postaci czytelnego rękopisu (dyskietki będą przez redakcję zwracane).
Objętość programu wraz z opisem i komentarzem nie powinna przekraczać 4 stron maszynopisu.
Raz w miesiącu Sąd Konkursowy wybierze najlepsze programy przyznając ich autorom dwie główne nagrody po

500.000 zł każda.

Decyzje Sądu Konkursowego są nieodwołalne. Oprócz zdobycia głównej nagrody autorzy mają szansę na publikację swych prac na łamach naszego pisma. Pracę prosimy podpisywać imieniem i nazwiskiem oraz dokładnym adresem autora.

Redakcja

WSZYSTKICH ZAINTERESOWANYCH NABYCIEM ZALEGŁYCH NUMERÓW

„64 plus 4 & AMIGA"

INFORMUJEMY, ŻE POSIADAMY JESZCZE OGRANICZONĄ ILOŚĆ NUMERÓW OD LISTOPADA 1990R. DO GRUDNIA 1991R.

ZAMÓWIENIA PROSIMY KIEROWAĆ NA ADRES :
Przedsiębiorstwo ABUK sp. z o.o.,
87-200 Wąbrzeźno,
ul. 1 Maja 33.
(Pod tym adresem mieści się dział kolportażu - tam też prosimy przesyłać wszelką korespondencję dotycząca kolportażu czasopisma, dyskietek, taśm itd. Adres redakcji się nie zmienił - patrz stopka.)

$68	104	bajt przekroczenia zakresu FAC1
$69-$6e	105-110	akumulator zmiennoprzecinkowy #2(FAC2)
$6f	111	flaga porównań przedznaków FAC 1,2
$70	112	pamięć zaokrągleń dla FAC 1,2
$71/$72	113/114	pamięć chwilowa dla procedur
$73-$8a	115-138	procedura CHRGET - pobranie kolejnego znaku
$8b-$8f	139-143	wskaźnik funkcji RND jako liczba zmiennoprzecinkowa
$90	144	status bajtu (ST) dla kanału wejścia/wyjścia
$91	145	pamięć chwilowa dla kontroli klawisza STOP (zawiera $7f, jeśli naciśnięty STOP)
$92	146	stała czasowa dla odczytu z taśmy
$93	147	flaga dla LOAD-(0) lub VERIFY-(1)
$94	148	flaga magistrali szeregowej
$95	149	następne znaki bufora wydawania
$96	150	flaga końca odczytanego bloku danych z taśmy
$97	151	pamięć chwilowa rejestru X
$98	152	liczba otwartych zbiorów
$99	153	numer aktualnego kanału wejścia
$9a	154	numer aktualnego kanału wyjścia
$9b	155	bajt kontrolny przy transmisji z magnetofonem
$9c	156	flaga prawidłowo odczytanego bajtu z taśmy
$9d	157	flaga dla programu (0), lub trybu bezpośredniego - (128)
$9e	158	błąd odczytu taśmy - przebieg 1
$9f	159	błąd odczytu taśmy - przebieg 2
$a0-$a2	160-162	wewnętrzny zegar TI i TI$
$a3	163	odczyt/zapis na taśmie - licznik bitów
$a4	164	odczyt/zapis na taśmie - licznik impulsów
$a5	165	licznik bitów synchronizacji taśmy
$a6	166	licznik bajtów w buforze magnetofonu
$a7	167	pamięć chwilowa dla operacji magnetofonowych i RS232
$a8	168	licznik bitów dla operacji magnetofonowych i RS232
$a9	169	flaga RS232 dla sprawdzania bitu startu
$aa	170	wprowadzenie RS232, pamięć chwilowa dla procedur magnetofonu
$ab	171	sprawdzenie prawidłowości otrzymywanych danych z RS232 i magnetofonu
$ac/$ad	172/173	adres startowy dla LOAD/SAVE, pamięć chwilowa dla SCREENEDITOR
$ae/$af	174/175	wskaźnik końca adresu dla LOAD/SAVE, pamięć chwilowa dla SCREENEDITOR
$b0/$b1	176/177	stała czasowa dla sterowania taktami taśmy
$b2/$b3	178/179	wskaźnik bufora kasety (zwykle 828)
$b4	180	licznik bitów RS232,
$b5	181	RS232 - wskazanie następnego bitu, flaga końca taśmy
$b6	182	RS232 - następny bit, flaga dla błędów odczytu taśmy
$b7	183	długość aktualnej nazwy zbiorów
$b8	184	aktualny logiczny numer zbioru
$b9	185	aktualny wtórny adres
$ba	186	aktualny numer kanału

C-64

$bb/$bc	187/188	wskaźnik aktualnej nazwy zbiorów
$bd	189	kontrola transmisji RS232, pamięć chwilowa dla operacji kasetowych
$be	190	licznik bloków dla zapisu i odczytu danych z taśmy
$bf	191	pamięć chwilowa dla LOAD z taśmy
$c0	192	flaga sterująca silnikiem magnetofonu
$c1/$c2	193/194	adres startowy dla LOAD i SAVE z taśmy
$c3/$c4	195/196	wskaźnik początku programu za nagłówkiem taśmy
$c5	197	kod ostatnio wciśniętego klawisza (64 - nie naciśnięto żadnego)
$c6	198	liczba znaków w buforze klawiatury
$c7	199	znaki: inwersja(18), normal (0)
$c8	200	licznik liczby znaków linii
$c9/$ca	201/202	pozycja kursora
$cb	203	kod klawisza
$cc	204	flaga trybu kursora (0=migający, 1=stały)
$cd	205	częstotliwość migotania kursora
$ce	206	kod ekranowy znaków pozycji kursora
$cf	207	flaga fazy migotania (1=kursor jest, 0=nie ma)
$d0	208	flaga wprowadzenia (0=klawiatura, 3=ekran)
$d1/$d2	209/210	początek aktualnej linii ekranowej
$d3	211	wartość X pozycji kursora w aktualnej linii
$d4	212	flaga trybu-limit (0=kursor przesuwa się 1=wydawanie znaków w inwersji, itp.)
$d5	213	długość aktualnej linii ekranowej
$d6	214	numer linii kursora
$d7	215	ostatni znak wydrukowany na ekranie
$d8	216	liczba znaków do wstawienia (po użyciu funkcji INST)
$d9-$f2	217-242	tabela starszych bajtów wskaźnika po rozpoczęciu linii ekranowej w RAM'ie (młodsze bajty znajdziemy w ROM'ie od adreseu $ecf0).
$f3/$f4	243/244	wskaźnik bajtu kolor-RAM'u, który należy do aktualnej linii ekranowej
$f5/$f6	245/246	adres aktualnej tabeli kodów klawiatury
$f7/$f8	247/248	RS232: wskaźnik początku bufora wejścia
$f9/$fa	249/250	RS323: wskaźnik początku bufora wyjścia
$fb-#fe	251-254	nie używane

Dalszy ciąg przedstawimy za miesiąc.

AMIGA

Na życzenie szerokiego grona czytelników (zwłaszcza tych początkujących) zamieszczamy dzisiaj opis programu

AsmOne.

AsmOne jest bardzo prostym w obsłudze „kombajnem" do pisania programów w asemblerze. Dlaczego „kombajn"? To proste - program ten zawiera w sobie bardzo dobry edytor, asembler, monitor oraz debugger czyli odpluskwiacz.

Pracę rozpoczyna się udzielając odpowiedzi na pytania dotyczące ilości obszaru roboczego jaki chcemy zapewnić dla naszego programu. Pamięć możemy przydzielić w obszarze Chip, Fast oraz pod adresem absolutnym. Ilość pamięci jaką przydzielimy zależy tylko od objętości projektu. Dla małych programów przydzielamy około 30-50 KB.

Gdy już przebrnęliśmy przez przydział pamięci możemy rozpoczynać pracę. Wszystkie funkcje są dostępne z menu (wciskamy prawy guzik myszki i wybieramy odpowiednią opcję) lub z klawiatury (poprzez wpisanie kilku liter z klawiatury lub poprzez wciśnięcie danego klawisza wraz z klawiszem "Amiga"("A")).

Przejście do edytora i wyjście z niego następuje za pomocą klawisza ESC. Za pomocą klawisza ESC wychodzimy również z funkcji monitora oraz debuggera.

Jeżeli nie wchodziliśmy do edytora ani innych funkcji to aktualnie znajdujemy się na ekranie roboczym i możemy wybierać funkcje z trzech głównych menu: Project, Assembler, Command. (Przy opisie komend w nawiasie są podane kombinacje klawiszy za pomocą których uzyskujemy te funkcje).

Menu Project:

Zap Source (ZS) - wykasowuje tekst źródłowy z edytora. Tekst nie jest usuwany z pamięci a jedynie edytor jest poinformowany o zerowej długości tekstu źródłowego.

Old (O) - przywraca skasowany tekst za pomocą komendy Zap Source (jeżeli np. skasowaliśmy go przypadkowo). Komenda ta podaje także wielkości obszarów roboczych.

Read - zestaw trzech komend służących do wczytywania danych do pamięci:

Read Source (R) - wczytuje tekst źródłowy do edytora.

Read Binary (RB) - wczytuje dane pod podany adres. Podajemy adres początku i końca obszaru przeznaczonego na wczytanie danych. Adres końca można pominąć. Tę funkcję należy stosować tylko wtedy gdy mamy pewność, że wczytując dane niczego nie uszkodzimy.

Read Object (RO) - wczytuje obiekt (gotowy uruchamialny program) do pamięci.

Write - zestaw czterech komend umożliwiających zapisywanie danych na dysku.

Write Source (W) - zapisuje tekst źrołowy z edytora na dysk.

Write Binary (WB) - zapisuje obszar danych na dysku. Podajemy początek i koniec tego obszaru.

Write Object (WO) - jeżeli piszemy program uruchamialny to po asemblacji możemy go nagrać na dysku za pomocą tej funkcji. (Niektóre z kradzionych wersji AsmOne'a mają tę funkcję uszkodzoną i chociaż zapisują obiekty na dysku to niestety później system nie rozpoznaje ich jako obiekty lecz jako zwykłe dane. Można taki obiekt poprawić, ale nie jest to proste).

Write Link (WL) - nagrywa na dysk zasemblowany program jako plik dołączalny.

Insert (I) - umożliwia dołączenie pliku tekstowego do tekstu źródłowego znajdującego się w edytorze. Plik dołączany jest na pozycji kursora w edytorze.

Update (U) - umożliwia zapisanie tekstu źródłowego z edytora. Jest to bardzo wygodna funkcja gdyż zapisuje ten tekst pod starą nazwą; wystarczy tylko wpisać "U" i tekst zostaje zapisany.

Zap File (ZF) - umożliwia skasowanie zbioru na dysku (dział jak funkcja Delete).

Preferences - umożliwia ustawienie preferencji:

Rescue - jeżeli funkcja ta jest włączona to po powrocie z programu zostaje ustawiony CopperList taki jak przed wejściem do programu.

Level 7 - umożliwia przerwanie działania jakiegokolwiek programu za pomocą klawisza ESC.

NumLock - po włączeniu tej funkcji, klawiatura numeryczna działa jako kursory (Home, End, etc.).

ReqLibrary - program używa req.library do wszelkiego typu requesterów, requesterów do zbiorów, etc.

Interlace - ekran roboczy jest uruchomiony w trybie Interlace (mamy dwa razy więcej linii widocznych w edytorze).

1 Bitplane - uruchamia ekran roboczy na jednym bitplanie (nie przyśpiesza to pracy a jedynie daje większą ilość pamięci do dyspozycji).

Source .S - program będzie wczytywał tylko zbiory z dopełnieniem .S.

CloseWB - zamyka WorkBench po wczytaniu programu.

WritePref (WP) - zapisuje preferencje na dysku (są one automatycznie odczytywane przy starcie programu).

About - notka o programie.

Exit AmOne (!) - wyjście z programu lub powtórne jego uruchomienie aby zmienić obszar roboczy.

Menu Assembler:

Assemble - zawiera opcje asemblacji i optymalizacji oraz preferencje dla asemblera:

Assemble ("A"+A lub A) - dokonuje asemblacji tekstu źródłowego zawartego w edytorze (asemblacja oraz optymalizacja mogą zostać przerwane za pomocą kombinacji CTRL+C).

Optimize ("A"+O) - dokonuje optymalizacji tekstu źródłowego oraz asemblacji. Optymalizacja polega na zastępowaniu dłuższych rozkazów krótszymi.

ListFile - jeżeli ta opcja jest włączona to przy asemblacji jest pokazywany tekst źródłowy oraz kody na jakie został zasemblowany.

Pagging - gdy włączona jest opcja ListFile to listing zostaje dzielony na strony.

HaltPage - gdy włączymy opcję ListFile i Pagging to po każdej stronie wylistowanej podczas asemblacji następuje wstrzymanie listowania i oczekiwanie na klawisz.

AllErrors - podczas asemblacji są pokazywane wszystkie błędy.

Debug - do zasemblowanego pliku uruchamialnego jest dołączany zbiór debug umożliwiający później łatwiejsze przeglądanie zbiorów za pomocą monitora lub debuggera.

Label: - gdy ta opcjá jest włączona to wszystkie etykiety muszą być zakończone dwukropkiem.

Ucase=LCase - wszystkie symbole mogą być pisane zarówno dużymi jak i małymi literami; wielkość liter nie odgrywa roli.

Editor ("A"+E) - przejście do edytora.

Debugger ("A"+D) - przejście do debuggera (najpierw następuje asemblacja przy włączonej opcji Debug).

Monitor ("A"+M) - przejście do monitora.

LineNumbers - gdy włączymy tę opcję to linie w edytorze mają numery.

AutoIndent - gdy pracujemy w edytorze to po wciśnięciu klawisza Return następuje przejście do następnej linii z jednoczesną tabulacją taką samą jak w linii poprzedniej.

Menu Command:

Editor - funkcje dotyczące edytora:

Top (T) - przejście na początek tekstu źródłowego.

Bottom (B) - przejście na koniec tekstu źródłowego.

Search (L) - poszukuje danego ciągu znaków w tekście źródłowym.

Zap Line (ZL) - kasuje jedną linię w tekście źródłowym a jeżeli po ZL wpiszemy ilość linii to kasuje podaną ilość linii.

PrintLine (P) - drukuje podaną ilość linii na drukrce.

Memory - funkcje operacji na pamięci:

Edit (M) - modyfikacja pamięci pod podanym adresem.

DisAssem (D) - disasemblacja pamięci od podanego adresu.

HexDump (H) - przeglądanie pamięci jako wartości w kodzie szesnastkowym.

Ascii (N) - przeglądanie pamięci jako kodów ASCII.

DisLine (@D) - disasemblacja pamięci w podanym zakresie (jako listing).

Assemble (@A) - bezpośrednia asemblacja do pamięci (pominięcie tekstu źródłowego).

HexLine (@H) - przeglądanie pamięci jako wartości szesnastkowych w podanym zakresie (jako listing).

AsciiLine (@N) - przeglądanie pamięci jako kodów Ascii w podanym zakresie (jako listing).

Search (S) - wyszukiwanie w pamięci pewnych wartości (podajemy obszar pamięci do przeszukiwania).

Fill (F) - wypełnienie pewnego obszaru pamięci podanymi wartościami (należy być ostrożnym z tą funkcją gdyż można bardzo łatwo wypełnić obszary zajmowane przez system).

Copy (C) - skopiowanie jednego obszaru pamięci w inne miejsce (należy zachować ostrożność przy kopiowaniu gdyż można uszkodzić system lub programy znajdujące się w pamięci).

Compare (Q) - porównuje zawartość jednego obszaru pamięci z innym i podaje komórki, w których wystąpiła niezgodność.

Insert - funkcje zawarte w tym menu umożliwiają dołączenie danych do tekstu źródłowego.

DisAssem (ID) - dane poddane zostają disasemblacji i dołączone do tekstu źródłowego.

HaxDump (IH) - dane zostaną dołączone jako wartości szesnastkowe.

Ascii (IN) - dane zostaną dołączone jako text Ascii.

Assemble - menu zawierające funkcje asemblacji:

Assemble (A) - asemblacja tekstu źródłowego.

Memory (@A) - asemblacja bezpośrednio do pamięci.

Optimize (AO) - optymalizacjia i asemblacja tekstu źródłowego.

Debug (AD) - przejście do debuggera.

Symbols (=S) - lista użytych symboli w zasemblowanym tekście źródłowym.

Monitor - funkcje dotyczące uruchamiania programu:

Jump (J) - skok do programu (działa jak JSR (GO SUB) czyli po zakończeniu programu następuje powrót, na stosie jest odłożony adres powrotu).

Go (G) - skok do programu (działa jak JMP (GO TO) czyli po zakończeniu wykonywania programu powrót nastąpi na wartość odłożoną na stosie czyli na ogół w zupełnie przypadkowe miejsce - po napotkaniu błednego rozkazu nastąpi powrót do AsmOne).

Step (K) - wykonywanie programu po jednej instrukcji.

Status (X) - podaje zawartośc rejestrów procesora (jeżeli podamy nazwę rejestru to możemy dokonać modyfikacji jego zawartości).

Zap BPS (ZB) - usuwa pułapki z programu.

Disk - funkcje obsługi dysku:

Read Sector (RS) - wczytanie sektora z dysku do pamięci.

Read Track (RT) - wczytanie ścieżki z dysku do pamięci

Write Sector (WS) - zapisanie sektora na dysku.

Write Track (WT) - zapisanie ścieżki na dysku.

CalcCheck (CC) - obliczenie sumy kontrolnej dla bloku.

Extern (E) - wykonanie wszystkich komend Extern znajdujących się w tekście źródłowym. Komenda Extern ma następującą składnię: EXTERN "nazwa zbioru", adres początku,adres końca - umożliwia ona załadowanie danych o podanej nazwie z dysku do danego obszaru pamięci.

Output () - skierowanie wyjścia danych na inne urządzenie niż ekran (na przykład na drukarkę czy do zbioru dyskowego).

Calculate (?) - oblicza wartość wyrażenia. Zamiast wartości liczbowych możemy podawać symbole lub etykiety.

I to już wszystkie funkcje dostępne z menu głównego. Po przejściu do edytora (za pomocą klawisza ESC) mamy do dyspozycji nowe menu.

Menu Edit Funct.:

Block - to menu dotyczy funkcji na bloku tekstu wyciętym z tekstu źródłowego:

Mark ("A"+b) - zaznaczenie początku bloku (w miejscu gdzie jest kursor).

Copy ("A"+c) - skopiowanie zaznaczonego bloku do bufora.

Cut ("A"+x) - wycięcie zaznaczonego bloku z tekstu źródłowego i przeniesienie go do bufora.

Insert ("A"+i) - dołączenie bloku znajdującego się w buforze na pozycji kursora.

Fill ("A"+f) - działanie jak Insert.

UnMark ("A"+u) - jeżeli przypadkowo zaznaczyliśmy w złym miejscu początek bloku to ta funkcja anuluje to zaznaczenie.

LowerCase ("A"+l) - wszystkie litery w bloku zostaną zamienione na małe.

UpperCase ("A"+L) - wszystkie litery w bloku zostaną zamienione na duże.

Rotate ("A"+y) - odwrócenie kolejności linii w zaznaczonym bloku.

Registers ("A"+k) - podaje rejestry używane w zaznaczonym bloku.

VerFill ("A"+n) - wstawienie bufora do tekstu źródłowego i przejście do następnej linii.

Search - to menu służy do wyszukiwania ciągów znaków w tekście:

Search ("A"+S) - wpisanie ciągu do poszukiwania oraz wyszukanie pierwszego ciągu.

Forward ("A"+s) - wyszukiwanie następnych ciągów.

Replace - to menu służy do zastępowania jednego ciągu znaków innym:

Replace ("A"+R) - wpisanie ciągów do wyszukania i do zamiany oraz wyszukanie pierwszego ciągu, który może być zmieniony. Pojawią się wtedy opcje Y/N/L/G - opcja Y - zamienia ciąg i wyszukuje następny, opcja N - pomija ciąg i wyszukuje następny, opcja L - zamienia ciąg i kończy wyszukiwanie, opcja G - zamienia wszystkie ciągi, które zostaną znalezione. Wciśnięcie jakiegokolwiek innego klawisza kończy wyszukiwanie.

Forward ("A"+r) - wyszukuje i zamienia ciąg znaków.

Del Line ("A"+d) - skasowanie linii i skopiowanie jej do bufora.

Marks - oznaczanie stałych pozycji w tekscie, do których można się będzie przenieść w dowolnym momencie (oznaczenia dokonujemy trzymając klawisz Amiga+Shift i wybierając klawisz od 1 do 3 natomiast skoku do takiej lokacji dokonujaemy trzymając klawisz Amiga i wybierając numer od 1 do 3).

Jump ;; ("A"+J) - dokonuje skoku do miejsca oznaczonego dwoma średnikami (jest to druga metoda przenoszenia się w dowolne miejsce tekstu źródłowego).

Jump Line ("A"+j) - wykonuje skok do podanej linii w tekście źródłowym.

Move - to menu zawiera funkcje przesuwania kursora. Kursor możemy przesuwać o całą stronę w górę lub w dół za pomocą klawiszy kursora góra/dół z jednocześnie wciśniętym klawiszem Shift. Przesunięcia do początku i końca linii dokonujemy poprzez wciśnięcie klawiszy kursora lewo/prawo wraz z klawiszem Shift. Klawisze kursora lewo/prawo wciśnięte wraz z klawiszem Alt przesuwają kursor na początek słowa następnego lub poprzedniego. Wciskając kombinacje Amiga i klawisz t przenosimy się na początek tekstu natomiast za pomocą kombinacji Amiga+Shift+t przenosimy się na koniec tekstu. Kombinacja Amiga+a przenosi kursor o 100 linii w górę a kombinacja Amiga+z przenosi kursor o 100 linii w dół.

MakeMacro ("A"+,) - funkcja ta umożliwia stworzenie makrobloku.

DoMacro ("A"+m) - wstawia makroblok na pozycji kursora.

I to wszystkie funkcje edytora. Po wciśnięciu kombinacji Amiga+M przechodzimy do monitora i uaktywnia nam się kolejne menu:

Mon Funct.:

DisAssem ("A"+d) - przejście do trybu disasemblacji.

HexDump ("A"+h) - przejście do trybu wyświetlania wartości szesnastkowo.

Ascii ("A"+n) - przejście do trybu przeglądania pamięci jako znaków Ascii.

Jump Addr ("A"+j) - ustawienie początku wyświetlania na podanym adresie.

Last Addr ("A"+l) - ustawienie początku wyświetlania na poprzednim adresie.

Mark i Jump - te funkcje działają tak jak w edytorze. Możemy oznaczyć dane adresy poprzez wciśnięcie kombinacji klawiszy Amiga+Shift oraz klawisz od 1 do 3 i skakać do tych oznaczonych lokacji poprzez wciśnięcie kombinacji Amiga z klaiwszem od 1 do 3.

QuickJump ("A"+q) - pobiera adres z danych i umożliwia skok pod ten adres jeżeli zatwierdzimy to wciśnięciem klawisza Return.

Only Ascii - jeżeli ustawimy tę opcję to podczas wyświetlania znaków Ascii będą wyświetlane tylko te które są rozpoznawane jako kody Ascii a więc od 32 do 127.

Kolejnym - zarazem ostatnim - jest menu debuggera.

Debug Funct.:

Step One (kursor w dół) - wykonywanie rozkazów po instrukcji tyle, że wszystkie skoki do procedur są wykonywane w całości.

Enter (kursor w prawo) - wejście do procedury, do której jest aktualnie wykonywany BSR lub JSR.

Run ("A"+r) - uruchomienie programu. Powrót do debuggera nastąpi gdy program się zakończy bądź podczas wykonywania napotka on pułapkę.

Step n ("A"+s) - wykonanie n instrukcji programu i przerwanie go.

Edit Regs ("A"+x) - umożliwia zmianę zawartości danego rejestru.

AddWatch ("A"+a) - umożliwia sprawdzanie jak zmienia się pamięć w czasi wykonywania programu. Pokazuje nam fragment pamięci wskazywany przez odpowiedni adres jako wartości lub znaki Ascii.

DelWatch - kasuje odpowiednio ustawione adresy do oglądania pamięci (kasowanie może nastąpić przez wciśnięcie klawisza Amiga z klawiszem od 1 do 8).

ZapWatch ("A"+Z) - usuwa wszystkie adresy, które ustawiliśmy funkcją AddWatch.

Jump ("A"+J) - umożliwia zmianę adresu, od którego następuje "odpluskwianie".

Jump mark ("A"+j) - umożliwia zmianę w małym zakresie. Dokonujemy tego przesuwając kursor na adres, który ma być początkowym i wciskając return.

B.P. Addr ("A"+B) - umożliwia ustawienie pułapki na podany adres. W miejscu, w którym jest ustawiona pułapka zakończy się wykonywanie programu i nastąpi powrót do debuggera.

B.P. Mark ("A"+b) - umożliwia ustawienie pułapki poprzez ustawienia kursora na danej pozycji i wciśnięcie klawisza return.

Zap all B.P. ("A"+z) - usuwa wszystkie ustawione wcześniej pułapki.

I to już wszystkie funkcje dostępne w programie AsmOne. Wszystkim, którym ten artykuł pomógł w obsłudze programu AsmOne życzę przyjemnej i owocnej pracy.

Marcin "Duddie" Dudar

REQTOOLS LIBRARY cz.1

AMIGA

Wraz z pojawieniem się systemu 2.0 rozpowszechnił się tzw. NewLook czyli graficzne przedstawianie elementów na ekranie w formie 3D (trójwymiarowo - czyli ze światłocieniami, w wypukłej bądź wklęsłej formie). Nico Francois, autor bardzo dobrego programu kompresującego dane, stworzył bibliotekę ReqTools, która umożliwia stosowanie w programach requesterów i file requesterów odpowiadających nowemu standardowi. Biblioteka jest wykonana w dwóch wersjach: dla Kickstartu 1.2/1.3 (około 25KB długości) oraz dla Kickstartu 2.0 (około 20KB długości). Niżej przedstawiony opis odnosi się do najnowszej wersji tej biblioteki v37.

Większość procedur tej biblioteki bazuje na taglist'ach czyli pewnych obszarach danych, które są utworzone w następujący sposób:

Kod Rozkazu 1 (długie słowo), Parametry (długie słowo)
Kod Rozkazu 2 (długie słowo), Parametry (długie słowo)
...
$00000000,$00000000

Wartości zerowe kończą taglist. Każda procedura obsługująca requester ma własne kody rozkazów ale istnieje pewna liczba rozkazów dostępnych ogólnie dla wszystkich requesterów.

rtAllocRequestA

Req = rtAllocRequestA (Typ, TagList)
D0 A0

Funkcja przydziela strukturę używaną przez funkcje obsługujące requestery (rtFileRequester, rtFontRequester, rtReqInfo). Struktura jest inicjowana automatycanie - aby użytkownik nie popełnił jakiegoś błędu, jednocześnie zapewnia to kompatybilność programu z innymi wersjami bibliotek. Z pamięci usuwamy ją za pomocą procedury rtFreeRequest().

Wejście:

Typ - typ struktury jaka ma być zainicjowana, aktualnie dostępne są typy RT_REQINFO (dla funkcji rtReqInfo), RT_FILEREQ (dla funkcji rtFileRequester), RT_FONTREQ (dla funkcji rtFontRequester).

TagList - wskaźnik listy rozkazów (aktualnie zero gdyż nie ma jeszcze rozkazów zdefiniowanych dla tej procedury).

Wyjście:

Req - wskaźnik struktury requestera przydzielonej przez tę funkcję lub zero jeżeli nie ma możliwości przydzielenia tej struktury (brak pamięci).

rtChangeReqAttrA

rtChangeReqAttrA (Req, TagList)
A1 A0

Funkcja zmienia atrybuty danej struktury requestera (na przykład nazwę katalogu, z którego ma korzystać funkcja rtFileRequester). Atrybuty są zmieniane na podstawie rozkazów przekazanych w liście rozkazów (TagList).

Wejście:

Req - wskaźnik struktury requestera, dla którego zmieniamy parametry.

TagList - wskaźnik tablicy rozkazów.

Dostępne rozkazy dla struktury requestera dla rtFileRequester:

RTFI_Dir - zmiana nazwy katalogu, który będzie odczytywany (jako parametr podajemy wskaźnik do zakończonego zerem ciągu znaków zawierającego nazwę). Bufor zawierający katalog zostanie opróżniony.

RTFI_MatchPat - zmiana wzoru, za pomocą którego dokonuje się sublimacji plików. Jako parametr podajemy wskaźnik tekstu zakończonego zerem zawierający ten wzór.

RTFI_AddEntry - dodanie pozycji do bufora zawierającego katalog. Ten rozkaz musi być ostatnim (od razu przed końcem listy rozkazów). Jako parametr podajemy strukturę uzyskaną z funkcji Dos.Lock() (musimy dokonać tego samodzielnie). Musimy się też upewnić czy pozycja, którą dodajemy znajduje się w katalogu znajdującym się w buforze. Jeżeli taka pozycja znajduje się już w katalogu to zostanie ona poprawiona.

RTFI_RemoveEntry - usuwa z katalogu znajdującego się w buforze zbiór o podanej nazwie. Jako parametr podajemy wskaźnik zakończonego zerem tekstu zawierającego nazwę.

Dostępne rozkazy dla struktury requestera dla rtFontRequester:

RTFO_FontName - ustawia nazwę aktualnie wybranej czcionki. Jako parametr podajemy zakończony zerem tekst zawierający nową nazwę.

RTFO_FontHeight - ustawia wysokość aktualnie wybranej czcionki. Jako parametr podajemy nową wysokość czcionki.

RTFO_FontStyle - ustawia styl czcionki. Styl podajemy jako parametr.

RTFO_FontFlags - ustawia znaczniki dla czcionki. Znaczniki podajemy jako parametr.

rtFreeReqBuffer

rtFreeReqBuffer (Req)
A1

Procedura dokonuje opróżnienia bufora dla danej struktury requestera (w przypadku requestera dla zbiorów zostanie wymazany katalog, natomiast w przypadku requestera dla czcionek zostanie wymazana ich lista). Można wywoływać tę funkcję nawet dla struktur requestera.

Wejście:

Req - wskaźnik struktury requestera, w której ma zostać opróżniony bufor.

rtFreeRequest

rtFreeRequest (Req)
A1

Zwalnia pamięć zajmowaną przez daną strukturę requestera (przydzieloną przez funkcję rtAllocRequest()). Funkcja zwalnia także bufor zajmowany przez dane w strukturze (nie ma potrzeby wywoływać funkcji rtFreeReqBuffer()). Po jej wykonaniu nie możemy już więcej używać tej struktury.

Wejście:

Req - wskaźnik struktury requestera, która ma być usunięta.

rtSetWaitPointer

rtSetWaitPointer (Window)
A0

Funkcja zmienia wygląd wskaźnika na tzw. wskaźnik oczekiwania.

Bardzo dobrze jest wywołać tę funkcje przed wywołaniem jakiegokolwiek requestera. Jeżeli użytkownik uaktywni okno przed opuszczeniem requestera pojawi się wskaźnik oznaczający zajęcie (użytkownik musi odpowiedzieć na requester przed zrobieniem czegokolwiek).

Wejście:

Window - wskaźnik struktury okna, dla którego ma być przypisany nowy wskaźnik

Marcin "Duddie" Dudar

OPIS ZESTAWU PUBLIC DOMAIN PACK

nr 17 (maj '92)

AMIGA

Chcąc wynagrodzić prenumeratorom Public Domain Pack'u, brak muzyczek w numerze marcowym i kwietniowym, na PDP-Maj'92, upchnąłem aż trzy moduły: [illegible]. Wszystkie są świetne i zasługują na wysoką lokatę w naszej biblioteczce PD.

A propos muzyczek to na majowym PD pack'u znajdziecie najnowszego playera do modułów - [illegible]. Jest to bardzo wygodny player z możliwością pracy w trybie "multi" czyli odtwarzania kilku muzyczek na zmianę. Wyboru modułu dokonujemy za pomocą requestera z req.library (szkoda, że autor nie posłużył się reqtools.library, wtedy wszystko wyglądało by jeszcze lepiej.). Tak więc do "podsłuchiwania" modułów mamy NoisePlayer'a, natomiast do układania własnych muzyczek posłuży nam [illegible], najnowszy program z rodziny Trackerów. Program ten został wyposażony w kilka nowych, bardzo ciekawych opcji (jak np. możliwość zamiany tekstu na sampl). Audio-Sculpture posiada ogromne możliwości obróbki sampli oraz tworzenia własnych sampli "od zera".

Następny to [illegible]. Jest to nowy program kopiujący, z bardzo ładną szatą graficzną, która nie psuje się po killsys'ie. Program posiada wiele ciekawych trybów kopiowania: trzy tryby dos'owe, trzy nibblery, a także opcja SAFE (to nie to samo co verify, które swoją drogą też jest). D-Copy posiada także opcje formatowania i testowania dysku. Po przetestowaniu nie trzeba już pamiętać co znaczy np. dwójka na czterdziestej piątej ścieżce (tak jak to jest w X-Copy), wystrczy kliknąć na ERR, następnie na ścieżkę, o której chcemy się czegoś dowiedzieć - i sprawa załatwiona; program sam wszystko grzecznie poda. Zaznacznie ścieżek do kopiowania jest również bardzo wygodne, lewym przyciskiem zaznaczamy początek, a prawym - koniec (na track display - czyli na dwóch siatkach kwadracików). Po killsys'ie nic się nie zmienia. W dalszym ciągu mamy do czynienia z przyjemną szatą graficzną. D-Copy posiada takżo możliwość zmiany SYNC'u czyli pozwala na kopiowanie w innych formatach. Przypominam, że SYNC jest skrótem od słów Synchronization Marking (znacznik synchronizacji) i jest to zbiór kilkunastu bitów (w naszym przypadku 16-tu), zapisanych na dysku aby oznaczyć początek zapisu danych. Skąd magiczna liczba 4489 (binarnie 0100010010001001)? Otóż jej zapis w formacie MFM używanym przez AmigaDOS jest niemożliwy. Dlaczego? Dlatego, że niemożliwe jest sąsiadowanie ze soba w tym formacie trzech zer. W formacie MFM po każdym bicie dodawany jest drugi bit (a więc dane zajmują dwa razy więcej miejsca niż powinny), jeżeli dwa sąsiadujące bity są zerami to dodawana jest jedynka, a jeżeli któryś z bitów jest jedynką to dodawane jest zero. Tak więc widać, że wygenerowanie trzech zer jest absolutnie niemożliwe.

Tak jak w X-Copy - przyciśnięcie prawego przycisku powoduje zablokowanie pointer'a. D-Copy posiada też opcję kompresji dysków, co rzadko zdarza się w programach kopiujących. Opcja ta będzie z pewnością bardzo przydatna osobom potrzebującym dużo miejsca na rzadko używane dane.



Na majowym Public Domain Pack'u znajdziecie także bardzo ciekawy program do kompresji dysków - [illegible]. Program służy tylko do kompresji dysków i ich archiwizacji (do jednego pliku), więc ma o wiele więcej opcji niż D-Copy i oczywiście polecałbym raczej DMS'a. Jest bardzo popularny we wszystkich BBS'ach krajowych i zagranicznych, więc na pewno może się przydać. DMS posiada wiele opcji dotyczących przepakowywania, przeglądania plików, pakowania dysków do jednego pliku i innych. Dla posiadaczy dysków twardych użycie DMS'a to jedyny sposób na przechowywanie "niedosowych" dysków. Jednak muszę zaznaczyć, że DMS nie jest programem Public Domain, jest to program shareware co oznacza, że aby stać się zarejestrowanym użytkownikiem należy wysłać pod adres: SDS Software, Suite 126, 4885-A McKnight Rd., Pittsburgh, PA 15237 - 30 dolarów za wersję podstawową lub 50 za wersję profesjonalną.

[illegible] można zakwalikować do programów typu CAD (Compuer Aided Design - czyli projektowanie wspomagane komputerem). Program służy do projektowania ogrodu bądź działki.

LandScape składa się z dwóch modułów. Pierwszy z nich pozwala na zaprojektowanie twojego ogrodu z uwzględnieniem takich detali jak ścieżki, dzewa, kwiaty, sadzawki i inne. Drugi z modułów wykonuje całą "czarną robotę", czyli tworzy dokładny rysunek na podstawie szkicu wykonanego przez użytkownika. Szkic jest rysowany pod dowolnym kątem i z dowolnej, wybranej przez nas odległości. Ogród tak narysowany (technika fraktali) wygląda jak prawdziwy. Aby jak najszybciej nauczyć się posługiwać programem radzę najpierw wykonać kilka prostych rysunków, a dopiero potem brać się za coś bardziej skomplikowanego (przecież rysowanie trwa dość długo, po co tracić czas bez potrzeby?). Aby wiedzieć co się rysuje podaję wymiary dostępnych w menu roślin:

Trees (Drzewa):
- dwarf deciduous tree (karłowate drzewo liściaste) - wys. ok. 1 m
- small deciduous tree (małe drzewo liściaste) - wys. ok. 2 m
- medium deciduous tree (średnie drzewo liściaste) - wys. ok. 5 m
- dwarf fir tree (karłowate drzewo iglaste) - wys. ok. 1 m
- small fir tree (małe drzewo iglaste) - wys. ok. 2 m
- medium fir tree (średnie drzewo iglaste) - wys. ok. 5 m

Non-flowering plants (czyli "zielsko"):
- small plant (mała roślinka) - wys. ok. 10 cm
- medium plant (średnia roślina) - wys. ok. 60 cm
- large plant (wielkie zielicho) - wys. ok. 150 cm
- small hedge element (kwałek małego żywopłotu)- wys. ok. 75 cm
- medium hedge element (kawał średniego żywopłotu) - wys. ok. 150 cm

Flowering plants (czyli kwiatki):
- small flower (kwiatek) - wysokość ok. 10 centymetrów
- medium flower (kwiacior) - wysokość ok. 60 centymetrów
- rose bush (róża) - wysokość ok. 60 centymetrów

Każdy kwiatek może być czerwony, pomarańczowy, żółty, niebieski lub kolor może być wybrany losowo przez komputer.

Ostatnia już pozycja z majowego Public Domain Pack'u to [illegible], program, który na pewno przyda się osobom zajętym i nie mającym czasu. Jest to komputerowy kalendarz, który umożliwia zapisanie terminu i czegoś co mamy w tym terminie do zrobienia. Pozwala na ustalenie priorytetu każdego zdarzenia i wybór jednego z dwóch requesterów, który nam pomże nie zapomnieć o jakiejś ważnej sprawie. Aby korzystać z programu, należy dopisać dwie linie do pliku Startup-Sequence naszego dysku systemowego:

assign dd: sys:katalog_w_którym_znajdują_się_programy
dd:diarycheck

oraz skopiować katalog DateDiary z PDP-Amiga. Wtedy po każdorazowym zbootowaniu dysku program diarycheck sprawdzi, czy nie mamy czegoś do zrobienia. Pożądany jest zegar systemowy (nie jest konieczny). Jeżeli go nie ma trzeba wpisywać za każdym razem datę z klawiatury.

Do zobaczenia za miesiąc!

Michał "Amber" Gosztyła

INCLUDE

czyli wartości, o których zapomnieliśmy

AMIGA

W kwietniowym numerze 64 PLUS 4 & AMIGA zaprezentowaliśmy dokumentację do PowerPacker Library, jednak z przyczyn nikomu niewiadomych nie znalazły się tam przesunięcia do odwołań do procedur względem rejestru zawierającego bazę biblioteki. Natychmiast zwróciliście nam w swoich listach uwagę na ten fakt. Naprawiamy nasz błąd i przepraszamy. Oprócz wartości dla PowerPacker Library prezentujemy także wartości dla ReqTools Library.

```
; PowerPacker Library
ppLoadData            equ -30
ppDecrunchBuffer      equ -36
ppCalcChecksum        equ -42
ppCalcPasskey         equ -48
ppDecrypt             equ -54
ppGetPassword         equ -60
ppAllocCrunchInfo     equ -96
ppFreeCrunchInfo      equ -102
ppCrunchBuffer        equ -108
ppWriteDataHeader     equ -114
ppEnterPassword       equ -120
ppErrorMessage        equ -132

DECR_COL0             equ 0
DECR_COL1             equ 1
DECR_POINTER          equ 2
DECR_SCROLL           equ 3
DECR_NONE             equ 4

PP_OPENERR            equ -1
PP_READERR            equ -2
PP_NOMEMORY           equ -3
PP_CRYPTED            equ -4
PP_PASSERR            equ -5
PP_UNKNOWNPP          equ -6
PP_EMPTYFILE          equ -7

SPEEDUP_BUFFLARGE     equ 0
SPEEDUP_BUFFMEDIUM    equ 1
SPEEDUP_BUFFSMALL     equ 2

CRUN_FAST             equ 0
CRUN_MEDIOCRE         equ 1
CRUN_GOOD             equ 2
CRUN_VERYGOOD         equ 3
CRUN_BEST             equ 4
```

```
; ReqTools Library
rtAllocRequestA             equ -30
rtFreeRequest               equ -36
rtFreeReqBuffer             equ -42
rtChangeReqAttrA            equ -48
rtFileRequestA              equ -54
rtFreeFileList              equ -60
rtEZRequestA                equ -66
rtGetStringA                equ -72
rtGetLongA                  equ -78
rtInternalGetPasswordA      equ -84
rtInternalEnterPasswordA    equ -90
rtFontRequestA              equ -96
rtPaletteRequestA           equ -102
rtReqHandlerA               equ -108
rtSetWaitPointer            equ -114
rtGetVScreenSize            equ -120
rtSetReqPosition            equ -126
rtSpread                    equ -132
rtScreenToFrontSafely       equ -138

RT_FILEREQ                  equ 0
RT_REQINFO                  equ 1
RT_FONTREQ                  equ 2
CALL_HANDLER                equ $80000000

RT_TagBase                  equ $80000000
RT_Window                   equ (RT_TagBase+1)
RT_IDCMPFlags               equ (RT_TagBase+2)
RT_ReqPos                   equ (RT_TagBase+3)
RT_LeftOffset               equ (RT_TagBase+4)
RT_TopOffset                equ (RT_TagBase+5)
RT_PubScrName               equ (RT_TagBase+6)
RT_Screen                   equ (RT_TagBase+7)
RT_ReqHandler               equ (RT_TagBase+8)
RT_DefaultFont              equ (RT_TagBase+9)
RT_WaitPointer              equ (RT_TagBase+10)
RTEZ_ReqTitle               equ (RT_TagBase+20)
RTEZ_Flags                  equ (RT_TagBase+22)
RTEZ_DefaultResponse        equ (RT_TagBase+23)
RTGL_Min                    equ (RT_TagBase+30)
RTGL_Max                    equ (RT_TagBase+31)
RTGL_Width                  equ (RT_TagBase+32)
RTGL_ShowDefault            equ (RT_TagBase+33)
RTGS_Width                  equ RTGL_Width
RTGS_AllowEmpty             equ (RT_TagBase+80)
RTFI_Flags                  equ (RT_TagBase+40)
RTFI_Height                 equ (RT_TagBase+41)
RTFI_OkText                 equ (RT_TagBase+42)
RTFO_Flags                  equ RTFI_Flags
RTFO_Height                 equ RTFI_Height
RTFO_OkText                 equ RTFI_OkText
RTFO_SampleHeight           equ (RT_TagBase+60)
RTFO_MinHeight              equ (RT_TagBase+61)
RTFO_MaxHeight              equ (RT_TagBase+62)
RTFI_Dir                    equ (RT_TagBase+50)
```

```
RTFI_MatchPat          equ
                       (RT_TagBase+51)
RTFI_AddEntry          equ
                       (RT_TagBase+52)
RTFI_RemoveEntry       equ
                       (RT_TagBase+53)
RTFO_FontName          equ
                       (RT_TagBase+63)
RTFO_FontHeight        equ
                       (RT_TagBase+64)
RTFO_FontStyle         equ
                       (RT_TagBase+65)
RTFO_FontFlags         equ
                       (RT_TagBase+66)
RTPA_Color             equ
                       (RT_TagBase+70)
RTRH_EndRequest        equ
                       (RT_TagBase+60)

REQPOS_POINTER         equ 0
REQPOS_CENTERWIN       equ 1
REQPOS_CENTERSCR       equ 2
REQPOS_TOPLEFTWIN      equ 3
REQPOS_TOPLEFTSCR      equ 4
REQ_CANCEL             equ 0
REQ_OK                 equ 1

FREQF_MULTISELECT      equ $1
FREQF_SAVE             equ $2
FREQF_NOBUFFER         equ $4
FREQF_NOFILES          equ $8
FREQF_PATGAD           equ $10
FREQF_FIXEDWIDTH       equ $20
FREQF_COLORFONTS       equ $40
FREQF_CHANGEPALETTE    equ $80
FREQF_LEAVEPALETTE     equ $100
FREQF_SCALE            equ $200
FREQF_STYLE            equ $400
FREQF_DOWILDFUNC       equ $800
FREQF_SELECTDIRS       equ $1000

EZREQF_NORETURNKEY     equ $1
EZREQF_LAMIGAQUAL      equ $2
EZREQF_CENTERTEXT      equ $4

REQHOOK_WILDFILE       equ 0
REQHOOK_WILDFONT       equ 1
```

Marcin "Duddie" Dudar

AMIGA

Pen Pal v 1.3

Ostatnio na rynku oprogramowania Amigi pojawiło się wiele bardzo skomplikowanych programów DTP (Desktop Publishing). Ich możliwości są duże, ale programy te nie należą do najłatwiejszych w obsłudze. Lukę pomiędzy skomplikowanymi programami DTP a prostymi edytorami tekstu wypełniają programy zwane word processorami.

Word processor to program posiadający pewne właściwości programów DTP, ale będący raczej edytorem. Do takiej właśnie grupy zalicza się Pen Pal v1.3.

Pen Pal to nie tylko word processor jakich jest wiele, Pen Pal to system, na który składa się oprócz edytora oraz całkiem niezła baza danych. Zastosowań dla takiego połączenia jest bardzo dużo i mogą one być bardzo ciekawe. Program umożliwia równoczesną pracę na czterech dokumentach, dzięki wbydowanemu słownikowi (zawiera ponad 100.000 słów), można sprawdzać poprawność pisowni (niestety angielskiej!). Bez żadnych problemów można importować grafikę w formacie IFF - zarówno do bazy danych jak i edytora. Dostępne jest automatyczne otoczenie dowolnej grafiki tekstem (tzw. text flow), przy użyciu dowolnego fontu kompatybilnego z Workbenchem (w tej chwili jest ponad 200 fontów), w różnych rozmiarach i kolorach.

Parametry bazy danych też nie są najgorsze - umożliwia ona operowanie na 32 polach i maksymalnie 32.000 rekordach na jedną bazę (można przecież otworzyć ich cztery!). Szybkość sortowania wynosi mniej niż pięć sekund dla bazy o wielkości 1000 rekordów (przy maksymalnej ilości pól).

Zalecane minimum do pracy z programem to 1 MB pamięci i zewnętrzna stacja dysków. Możliwa jest praca z jedną stacją dysków, ale jest to dosyć męczące.

Co prawda program został stworzony z myślą o komforcie i łatwości obsługi, ale myślę, że warto dokładnie poznać jego tajniki aby móc maksymalnie wykorzystać jego walory.

Oto opis wszystkich opcji programu:

SYSTEM:

Status - zbiór bardzo przydatnych informacji o ilości wolnej pamięci, ilości pamięci używanej przez program a także, co bardzo ważne, informacje o wolnym miejscu na dysku. Zawsze warto mieć czysty dysk na swoje dane na dysku z programem jest mało miejsca, poza tym zapisywanie danych na tym samym dysku, na którym jest program to nie najlepsze wyjście.



Preferences - czyli rozmaite opcje programu (uruchamianie w zapisanej na dysku konfiguracji), dostępne są Program, Document, Data Base, Text Print, Graphic Print. Każda z tych opcji dotyczy innej części programu.

Help - wiele pożytecznych wskazówek dotyczących obsługi programu.

About Pen Pal - zwięzła informacja o twórcach Pan Pal'a.

FILE:

Windows - umożliwia przeglądanie projektów, gdy pracujemy nad kilkoma jednocześnie.

New - tu do wyboru jest otwarcie nowego dokumentu (New Document) lub otwarcie nowej bazy danych (New Data Base).

Open - załadowanie uprzednio zapisanej bazy danych lub dokumentu.

Close - zamknięcie projektu (program zapyta czy chcemy sobie go zapisać).

Save - zapisuje bazę danych lub dokument do aktualnego pliku.

Save As - tworzy nowy plik (o nazwie podanej przez użytkownika) i do niego zapisuje efekty naszej pracy.

Print - czyli wydrukowanie projektu. Prosi o podanie takich parametrów jak ilość kopii, rodzaj grafiki i fontów.

Rename - zmiana nazwy istniejącego już na dysku pliku z projektem.

Delete - usunięcie pliku.

Quit - zakończenie pracy z programem. Wyjście do AmigaDOS.

Program jest zabezpieczony przed nieuwagą użytkownika i zanim stracisz sześć godzin pracy program zapyta czy przypadkiem nie chcesz zapisać projektu na dysku. Jeżeli chcesz to kliknij na YES, jeżeli nie to użyj NO.

Opcja CANCEL służy do wycofania się, program potraktuje wszystko tak jak gdyby opcja QUIT nie została użyta.

Przy zapisie pliku program wyświetla requester, dzięki któremu wygodnie wybieramy potrzebny dysk oraz ścieżkę i pyta w jakim formacie ma zapisać dane. Dostępne są dwie możliwości: format PP (format używany wyłącznie przez Pen Pal'a), oraz popularny format ASCII. Polecam ten drugi ze względu na kompatybilność z masą rozmaitych programów (od zwykłych "przeglądaczek" do potężnych programów typu DTP).

EDIT:

Undo - naprawia nasze "nieodwracalne" błędy. Należy spróbować użyć tej opcji gdy coś przypadkiem skasujesz (grafikę lub tekst, bo ta sztuczka nie uda się ze skasowanym plikiem!).

Cut - usuwa tekst (oczywiście komputer musi wiedzieć co usunąć). Tekst zaznacza się przy pomocy opcji Select.

Paste - powoduje skopiowanie zaznaczonego Select'em tekstu do bufora.

Copy - skopiowanie tekstu z bufora do miejsca na ekranie, które wskazuje kursor.

Insert Page - dodanie strony do dokumentu.

Insert Date - wprowadza datę (jeżeli masz zegar).

SEARCH:

Search/Replace - służy do zastępowania podanego ciągu znaków innym. Ciągi mogą być różnej długości. Np. możemy zastąpić dwie litery - dwiema cyframi i literą itd. Ciąg, który chcemy usunąć podajemy po Search For, a ciąg którym chcemy go zastąpić - po Replace With.

Go to insertion point - służy do wskazywania miejsca umieszczenia nowego tekstu.

PROCESS:

Spell - opcja ta służy do sprawdzania poprawności pisowni słów angielskich.

Statistics - podaje szczegółowe informacje o tekście (ilość słów, ilość liter, zdań, spacji, rozdziałów etc.).

Page Setup - czyli przygotowanie wydruku dokumentu. Ustalamy ilość stron, rodzaj grafiki itd.

Options - posiadają następujące submenu:

Full page view - czyli podglad drukowanego tekstu w pomniejszeniu.

Hide rulers - służy do usunięcia podziałki u góry i po lewej stronie. Wadą programu jest fakt, iż podziałka jest w calach i nie ma sposobu aby zmienić ją na bardziej swojskie centymetry. Program został napisany w USA, gdzie obowiązują inne miary, ale to wcale nie usprawiedliwia autorów.

Hide bottom scroll bar - usuwa dolną belkę. Opcje te są zdublowane jako gadżety po prawej stronie każdego otwartego okna, nad nimi znajduje się jeszcze gadżet Full Window, który rozszerza okno do maksymalnych rozmiarów.

Define colors - pozwala na zmianę palety kolorów. Bardzo ciekawe jest to, że program ma przygotowane kilkanaście niezłych palet.

GRAPHICS:

Import picture - importowanie grafiki w formacie IFF do aktualnego projektu. Na dysku Extras znajduje się przykładowy plik IFF - "Bike" (w katalogu Pictures).

Text flow - duża zaleta Pen Pal'a - automatyczne układanie tekstu dookoła dowolnego obrazka.

Text - zawiera typy dostępnych czcionek. W tym menu możemy również wybrać rodzaj pisma (pogrubiony, pochylony, podkreślony itd.)

Text Color - pozwala na zmienianie koloru liter pojawiających się na ekranie.

SuperScript - unosi częściowo drukowaną linię.

SubScript - opuszcza drukowaną linię.

Text Aligment - mamy tu kilka opcji:

LEFT - rozpoczęcie druku od lewego marginesu.

RIGHT - rozpoczęcie druku od prawego marginesu.

CENTER - druk od srodka strony.

Line Spacing - umożliwia zmianę odległości pomiędzy kolejnymi wierszami tekstu.

SIZE:

To menu służy do wybierania rozmiaru fontów.

FONT:

Służy do wybierania rodzaju fontów. W każdej z tych opcji jako submenu ukaże się lista dostępnych fontów.

AMIGA

Ikony po lewej stronie (od góry do dołu):

1. Edytor tekstu - pozwala na wprowadzanie, kasowanie i wszystkie operacje tekstowe w wybranym dokumencie.
2. Opróżnienie zawartości zaprojektowanej ramki.
3. Wypełnienie ramki tekstem.
4. Pionowa kreska umożliwia rysowanie pionowych linii.
5. Pozioma, poziomych.
6. Strzałka - wybieranie (wskazywanie) grafiki do dalszej obróbki.
7. Przekreślony kwadracik - manipulowanie grafiką.
8. Wybieranie koloru linii.
9. Wybieranie koloru wypełnienia linii.

Kwadrat umieszczony nad paletą z kolorami umożliwia barwienie wnętrza ramki. Pod paletą z kolorami znajduje się regulator grubości linii. Aby wybrać np. kolor czerwony jako kolor linii trzeba kliknąć na 8 a następnie na kolor czerwony na palecie kolorów pod spodem. Jeżeli linia ma być grubsza a chcesz aby calutka była czerwona, należy powtórzyć operacje z ikoną 9.

Myślę, że to wystarczy aby jako tako zaznajomić się z możliwościami programu. Aby wiedzieć na jego temat nieco więcej potrzeba nieco praktyki, do której gorąco zachęcam.

Michał "Amber" Gosztyła

SpeedBall II - savegame

Po uważnym zbadaniu boot-blocku dysku, okazało si, że aby uzbrojenie wszystkich zawodników było maksymalne (tj. 250) wystarczy wykonać poniższą procedurę:

- uruchomić SpeedBall'a,
- zgrać save-game,
- przejrzeć boot-block dysku za pomocą jakiegoś dysk-monitora,
- zmienić powtarzające się wartości 100 (heksadecymalnie 64) na 250 (heksadecymalnie FA),
- uruchomić SpeedBall'a i wgrać uprzednio zgraną grę.

Oczywiście na boot-block'u znajduje się także licznik pieniędzy oraz cała tabela ligowa, ale to już musicie znaleźć sami.

AMBER

AMIGA

KĄCIK POCZĄTKUJĄCEGO KODERA

CZ.14

DCMP

IDCMP (Intuition's Direct Comunication Messages Ports) czyli porty bezpośredniej komunikacji biblioteki intuition z programami są najlepszą metodą otrzymywania komunikatów z biblioteki intuition w związku z sytuacjami, które zaistniały w komputerze, na przykład wybranie danego menu, wciśnięcie gadżetu, wciśnięcie przycisku myszki czy klawisza na klawiaturze lub choćby włożenie albo wyjęcie dysku ze stacji. W przypadku zaistnienia takiej specjalnej sytuacji zostaje automatycznie wygenerowany komunikat dotyczący tego zdarzenia. Następnie biblioteka intuition sprawdza czy program jest zainteresowany w otrzymaniu komunikatu na ten temat i ewentualnie wysyła go do naszego programu. Każde okno zawiera listę znaczników IDCMP, określającą zdarzenia o jakich program, w którym to jest okno uruchomione, chce być informowany. Jeżeli program jest zainteresowany danym komunikatem otrzyma go w formie struktury IntuiMessage.

Niektóre ze zdarzeń są ważne dla jednego programu, na przykład wykrycie znacznika GADGETDOWN odpowiadające wciśnięciu gadżetu w obrębie aktywnego okna i zostaną przekazane tylko do tego programu. Natomiast niektóre ze zdarzeń są ważne dla różnych programów, na przykład wykrycie znacznika DISKINSERTED odpowiadające włożeniu dysku do stacji - będą one przekazane dla wszystkich programów zainteresowanych tym zdarzeniem.

Jak otrzymywać komunikaty IDCMP? Aby otrzymać komunikat IDCMP należy:

1. Otworzyć port IDCMP.
2. Poczekać na komunikat.
3. Zebrać komunikaty.
4. Zbadać komunikaty.
5. Odpowiedzieć na komunikat. (Należy poinformować bibliotekę intuition, że odebraliśmy dany komunikat gdyż w innym przypadku będzie on stale zajmował miejsce w pamięci).

Na komunikaty należy odpowiadać jak najszybciej jednak przed odpowiedzią należy zbadać czego dotyczy komunikat gdyż po odpowiedzi następuje jego automatyczne wymazanie. Najlepsza metoda to odebrać dany komunikat, przepisać go w całości w inne miejsce, odpowiedzieć i dopiero teraz przebadać i zadziałać w sposób odpowiedni do danego komunikatu.

Otwarcie portu IDCMP.

Port IDCMP jest automatycznie kreowany przez bibliotekę intuition w momencie gdy otwieramy okno (każde okno ma własny port komunikacji za pomocą znaczników IDCMP). Gdy otwieramy okno możemy zdefiniować w polu znaczników IDCMP w strukturze NewWindow które z komunikatów IDCMP chcemy otrzymywać. Jeżeli na przykład otworzymy okno z ustawionym znacznikiem GADGETDOWN to zostanie utworzony port IDCMP i nasz program będzie mógł odbierać komunikaty za każdym razem jeżeli tylko zostanie wciśnięty jakikolwiek gadżet w obrębie tego okna. Jeżeli już mamy otwarte okno to w każdej chwili możemy dokonać zmian w obrębie znaczników IDCMP dotyczących komunikatów jakie chcemy otrzymywać. Dokonujemy tego za pomocą procedury Intuition.ModifyIDCMP().

Oczekiwanie na komunikaty.

Kiedy zostało otwarte okno i wykreowany port komunikacji nasz program musi tylko oczekiwać na komunikaty. Istnieją dwa różne sposoby oczekiwania na komunikaty:

1. Sposób biemy. Nasz program jest zatrzymany i zostaje uaktywniony tylko wtedy gdy zostaje przekazany komunikat. Zwiększa to prędkość innych programów współdziałających z naszym na multitaskingu gdyż procesor nie zajmuje się naszym programem do czasu gdy nie zostanie mu przekazany komunikat. Do tego celu używa się funkcji Exec.Wait().

2. Sposób czynny. Nasz program stara się zbierać komunikaty i jeżeli nie otrzymuje żadnego ponawia tę próbę. Sposób ten jest niezbędny gdy program musi kontynuować wykonywanie jakiejś czynności i jednocześnie odbierać komunikaty. Oczekiwania na komunikaty dokonujemy za pomocą funkcji Exec.WaitPort().

Zbieranie komunikatów.

Aby odebrać komunikat od biblioteki intuition przez port IDCMP powinniśmy użyć funkcji Exec.GetMsg(). Funkcja ta zwraca wskaźnik struktury IntuiMessage, która zawiera wszelkie niezbędne informacje dotyczące komunikatu. W przypadku gdy otrzymamy zero, będzie to oznaczało, iż żaden komunikat nie jest kierowany do naszego programu.

Odpowiadanie na komunikaty.

Kiedy nasz program odczyta wszystko na temat danego komunikatu ze struktury IntuiMessage powinien dać odpowiedź do biblioteki intuition poprzez wywołanie funkcji Exec.ReplyMsg(). Informuje to bibliotekę intuition, iż zakończyliśmy odczytywanie komunikatu i może go ona wymazać. Po odpowiedzi na komunikat poprzez wywołanie funkcji Exec.ReplyMsg() nie możemy więcej zmieniać ani odczytywać struktury IntuiMessage. Inny program może ją w tym czasie już wykorzystywać lub może ona ulegać zmianie przez bibliotekę intuition.

Badanie komunikatów.

Jeżeli odebraliśmy komunikat, to należy zbadać strukturę IntuiMessage. Struktura IntuiMessage wygląda następująco:

```
struct IntuiMessage
STRUCT  Message      ; struktura ExecMessage
ULONG   Class
UWORD   Code
UWORD   Qualifier
APTR    IAddress
WORD    MouseX,MouseY
ULONG   Seconds,Micros
APTR    Window
APTR    SpecialLink  ; wskaźnik specjalnej struktury
                       IntuiMessage
```

A teraz opis poszczególnych elementów struktury IntuiMessage.

ExecMessage - struktura używana przez bibliotekę exec (nie należy tej struktury modyfikować).

Class - zawiera znacznik IDCMP (typ komunikatu) dla jakiego jest przeznaczony komunikat zawarty w strukturze IntuiMessage.

Code - zawiera różne specjalne wartości, które pozostają w bliskiej relacji ze znacznikiem IDCMP, dla którego jest generowany komunikat. Na przykład: jeżeli zostanie zwrócony w polu Class znacznik RAWKEY występujący wtedy gdy został naciśnięty klawisz podczas gdy okno było aktywne, to w polu Code znajdzie sie kod tego klawisza.

Qualifier - jeżeli nasz program otrzymał komunikat RAWKEY (kody klawiszy) to pole to zawiera informacje dotyczące stanu klawiszy Shift, Ctrl, Alt, etc. (jest to kopia pola ie_Qualifier).

MouseX - pozioma pozycja kursora myszki względem górnego lewego rogu okna.

MouseY - pionowa pozycja kursora myszki względem górnego lewego rogu okna.

Seconds - kopia pola sekund z zegara systemowego oznaczająca ilość sekund jakie upłynęły od 1 stycznia 1978 roku.

Micros - kopia pola mikrosekund z zegara systemowego.

IAddress - wskaźnik do obiektu, który wygenerował komunikat. Na przykład jeżeli mamy ustawiony znacznik GADGETDOWN i otrzymujemy komunikat GADGETDOWN (czyli wciśnięty gadżet) to pole to będzie zawierało adres struktury Gadget dla gadżetu, który wygenerował ten komunikat.

Window - wskaźnik struktury Window dla okna, do którego jest ten komunikat przeznaczony.

SpecialLink - wskaźnik używany przez biblioteki exec i intuition nie wolno go modyfikować.

Najlepszy sposób aby zbadać strukturę IntuiMessage otrzymaną jako komunikat to skopiować wszystkie ważne wartości, następnie odpowiedzieć tak szybko jak to tylko możliwe gdyż przetrzymywanie komunikatów bardzo spowalnia działanie biblioteki intuition. Po udzieleniu odpowiedzi na komunikat możemy się zająć skopiowanymi wartościami. (Należy pamiętać, aby nie używać więcej struktury IntuiMessage).

Typy znaczników IDCMP:

Znaczniki powiązane z gadżetami:

GADGETDOWN - nasz program otrzyma ten komunikat jeżeli wcisnęliśmy dany gadżet (gadżet musi mieć ustawiony znacznik GADGIMMEDIATE w polu Activation (na temat gadżetów będzie poświęcony kolejny odcinek Kącika Początkującego Kodera), jeżeli nie będzie on ustawiony to nie otrzymamy żadnego komunikatu).

GADGETUP - program otrzyma ten komunikat jeżeli gadżet został zwolniony (gadżet musi mieć ustawiony znacznik RELVERIFY w polu Activation gdyż inaczej nie będzie generowany komunikat GADGETUP).

CLOSEWINDOW - program otrzyma ten komunikat jeżeli użytkownik wybrał gadżet systemowy Close Window odpowiadający za zamknięcie okna. Należy pamiętać, iż okno nie będzie zamykane automatycznie i zależy to tylko od naszego programu czy okno zostanie zamknięte czy też pozostanie otwarte.

Znaczniki powiązane z oknem:

NEWSIZE - nasz program otrzyma ten komunikat jeżeli użytkownik zmienił wymiary okna. Aby znaleźć nowe wymiary okna należy sprawdzić 35 strukturę Window dla danego okna i zawarte w niej pola Width i Height (szerokość i wysokość).

SIZEVERIFY - nasz program otrzyma ten komunikat gdy użytkownik będzie próbował zmienić wymiary okna. Jakkolwiek, biblioteka intuition zmieni wymiary okna to jednak wcześniej wyśle ten komunikat co pozwoli nam na reakcję zanim okno zmieni wymiary.

REFRESHWINDOW - nasz program otrzyma ten komunikat jeżeli okno będzie wymagało odświeżenia. Odnosi się to tylko do okien typu SIMPLE_REFRESH i SMART_REFRESH. Należy pamiętać, że jeżeli otrzymamy komunikat REFRESHWINDOW to musimy wywołać procedury Intuiotion.BeginRefresh() i Intuition.EndRefresh() nawet jeżeli nie będziemy niczego odświeżać (dokładny opis znajduje się w Kąciku Początkującego Kodera z poprzedniego numeru dotyczącego okien).

ACTIVEWINDOW - nasz program otrzyma ten komunikat jeżeli okno zostało uaktywnione.

INACTIVEWINDOW - program otrzyma ten komunikat jeżeli okno przestało być aktywne.

Znaczniki powiązane z requesterami:

REQSET - program otrzyma ten komunikat jeżeli został otworzony requester w naszym oknie. Jest to bardzo użyteczna metoda sprawdzania czy w oknie pojawił się requester typu Double-Menu. (Komunikat otrzymamy zarówno wtedy gdy pojawi się ten requester jak requester normalny).

REQCLEAR - nasz program otrzyma ten komunikat jeżeli został usunięty ostatni requester powiązany z naszym oknem.

REQVERIFY - program otrzyma ten komunikat jeżeli ma być otworzony requester powiązany z naszym oknem. Jakkolwiek requester będzie otwarty po tym jak odpowiemy na komunikat, mamy jednak możliwość zakończyć pewne prace zanim zostanie on uaktywniony. Nasz program otrzymuje od biblioteki intuition komunikat tylko wtedy jeżeli jest otwierany pierwszy requester w oknie. O pojawianiu się requesterów intuition nie informuje gdyż przyjmuje, iż zakończyliśmy wszelkie prace z oknem i oczekujemy na pojawienie się komunikatu REQCLEAR.

Na temat requesterów będzie poświęcony artykuł w jednym z następnych numerów '64 PLUS 4' w Kąciku Początkującego Kodera.

Znaczniki powiązane z menu:

MENUPICK - program otrzyma ten komunikat jeżeli użytkownik wcisnął prawy przycisk myszki. Nie oznacza to jednakże, iż została wybrana jakaś pozycja z menu. Pole Code w strukturze IntuiMessage zawiera numer wybranego menu. Jeżeli nie została wybrana żadna pozycja z menu to komórka ta zawiera wartość MENUNULL. Na temat menu, jak je odczytywać, sprawdzać, które menu zostało wybrane będziemy pisać w Kąciku Początkującego Kodera.

MENUVERIFY - program otrzyma ten komunikat jeżeli menu ma być otwarte. Tu również mamy czas aby zakończyć prace i dopiero wtedy odpowiedzieć na komunikat. Jest to bardzo użyteczne zwłaszcza przy oknach, które odświeżamy samodzielnie gdyż przed otwarciem menu możemy dokonać niezbędnych czynności i zawartość okna nie ulegnie uszkodzeniu. Należy sprawdzić co zawiera pole Code

w strukturze IntuiMessage. Jeżeli jest to wartość odpowiadająca MENUHOT to oznacza, iż będzie otwarte menu należące do tego okna natomiast MENUWAITING oznacza, że zostanie wyświetlone menu należące do innego okna. Możemy zapobiec wyświetleniu menu poprzez zmianę pola Code w strukturze IntuiMessage na wartość odpowiadającą MENUCANCEL przed tym jak odpowiemy na komunikat.

Znaczniki powiązane z myszką:

MOUSEBUTTONS - nasz program otrzyma ten komunikat jeżeli jakikolwiek z przycisków myszki został wciśnięty lub zwolniony. Należy zbadać pole Code w strukturze IntuiMessage i sprawdzić jakiej wartości jest równe:

SELECTDOWN - lewy przycisk myszki został naciśnięty
SELECTUP - lewy przycisk myszki został puszczony
MENUDOWN - prawy przycisk myszki został naciśnięty
MENUUP - prawy przycisk myszki został puszczony

Jeżeli użytkownik wciska lewy przycisk myszki podczas gdy wskaźnik znajduje się w obrębie gadżetu, to nasz program nie otrzyma żadnego komunikatu dotyczącego wciśnięcia przycisku myszki. Zostaną wtedy wygenerowane komunikaty dotyczące gadżetów. Taka sama sytuacja może zaistnieć gdy wciskamy prawy przycisk myszki na pasku menu. Zostanie wyświetlone menu i zostaną wygenerowane komunikaty dotyczące menu, nie zostanie wygenerowany komunikat MENUDOWN dotyczący prawego przycisku myszki. Jeżeli nie chcemy aby pojawiało się menu i blokowało prawy przycisk myszki to musimy ustawić znacznik RMBTRAP w polu Flags w strukturze NewWindow.

MOUSEMOVE - program otrzyma ten komunikat jeżeli wskaźnik kursora myszki został przesunięty. Nasze okno musi mieć ustawiony znacznik REPORTMOUSE w polu Flags w strukturze NewWindow lub gadżet w oknie musi mieć ustawiony znacznik FOLLOWMOUSE w polu Activation w strukturze Gadget. Należy być przygotowanym na odbieranie bardzo dużej ilości komunikatów.

DELTAMOVE - działa tak samo jak MOUSEMOVE jednak z tą różnicą, iż przesunięcia będą przekazywane jako delty czyli różnice pomiędzy pozycją poprzednią a aktualną. Nawet jeżeli wskaźnik osiągnął koniec ekranu i się nie przesuwa to nasz program będzie nadal otrzymywał komunikaty o przesunieciu. Podobnie jak dla MOUSEMOVE będziemy otrzymywać bardzo duże ilości komunikatów.

Pozostałe znaczniki:

RAWKEY - program otrzyma ten komunikat za każdym razem gdy użytkownik wciśnie jakiś klawisz. Pole Code w strukturze IntuiMessage zawiera kod wciśniętego klawisza, natomiast pole Qualifier w strukturze IntuiMessage zawiera wskaźniki dotyczące klawiszy specjalnych jak Shift, Ctrl, etc.

VANILLAKEY - program otrzyma ten komunikat za każdym razem gdy użytkownik wciśnie klawisz. Pole Code w strukturze IntuiMessage zawiera kod ASCII wciśniętego klawisza.

DISKINSERTED - program otrzyma ten komunikat jeżeli dysk został włożony do stacji dysków. Otrzymają go wszystkie programy zainteresowane tym komunikatem.

DISKREMOVED - program otrzyma ten komunikat jeżeli dysk został wyjęty ze stacji dysków. Otrzymają go wszystkie programy zainteresowane tym komunikatem.

NEWPREFS - program otrzyma ten komunikat jeżeli zostały zmienione preferencje systemowe. Należy wywołać funkcję Intuition.GetPrefs() aby osiągnąć nowe preferencje. Na temat preferencji będziemy pisać w Kąciku Początkującego Kodera.

INTUITICKS - program otrzyma ten komunikat dziesięć razy na sekundę (około), jeżeli nie odpowiemy na jeden z tych komunikatów to wysyłanie ich zostanie wstrzymane.

WBENCHMESSAGE - program otrzyma ten komunikat jeżeli jakikolwiek program próbował zamknąć lub otworzyć WorkBench poprzez użycie funkcji Intuition.OpenWorkBench() lub Intuition.CloseWorkBench(). Pole Code w strukturze IntuiMessage zawiera wartość odpowiadającą WEBENCHOPEN jeżeli WorkBench został otworzony lub WBENCHCLOSE jeżeli WorkBench został zamknięty.

Funkcje:

GetMsg() - Exec

```
Komunikat - GetMsg ( Port )
  D0           A0
```

Funkcja ta zajmuje się pobraniem komunikatu z portu komunikacji.

Wejście:

Port - jako port komunikacji podajemy port UserPort, którego adres pobieramy ze struktury Window dla danego okna.

Wyjście:

Komunikat - jeżeli zero to nie otrzymaliśmy żadnego komunikatu, w przeciwnym przypadku jest to adres struktury IntuiMessage odpowiadającej za ten komunikat.

ReplyMsg() - Exec

```
ReplyMsg(Komunikat)
           A1
```

Funkcję tę wywołujemy gdy chcemy odpowiedzieć na dany komunikat po tym jak dokonaliśmy jego analizy gdyż później nie będzie już dostępny.

Wejście:

Komunikat - wskaźnik do struktury IntuiMessage jaki otrzymaliśmy z procedury Exec.GetMsg().

ModifyIDCMP() - Intuition

```
ModifyIDCMP(Window, IDCMPFlags)
              A0       D0
```

Funkcja zmienia znaczniki IDCMP ustawione dla danego okna w celu odbierania innych komunikatów niż zdefiniowane, bądź zablokowania odbierania pewnych komunikatów.

Wejście:

Window - wskaźnik struktury Window dla otwartego okna

IDCMPFlags - (długie słowo) żaden lub kilka znaczników IDCMP, które zostaną ustawione (poprzednie wartości zostaną wymazane).

I to już wszystko w tym miesiącu. W przyszłym Kąciku Początkującego Kodera informacje na temat gadżetów.

Marcin "Duddie" Dudar